(NASA-CR-186827) THE 1988 YEAR END REPORT N90-70880
ON AUTONOMOUS PLANETARY ROVER AT CARNEGIE
MELLON (Carnegie-Mellon Univ.) 94 p
Unclas
00/37 0293163

Carnegie Mellon University

The Robotics Institute

Technical Report

1988 Year End Report
Autonomous Planetary Rover at Carnegie Mellon

Takeo Kanade Tom Mitchell William Whittaker
Principal Investigators

CMU-RI-TR-89-3

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

January 1989

LIBRARY COPY
OCT 19 1989
LANGLEY RESEARCH CENTER
LIBRARY NASA
HAMPTON, VIRGINIA

©1989 Carnegie Mellon University

This research was sponsored by NASA under Contract NAGW-1175. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agency.

Contents

I Introduction 1
Overview 1
Accomplishments 4
Personnel 6
Publications 6

II An Autonomous Rover for Exploring Mars 9

III Configuration of an Autonomous Robot for Mars Exploration 43

IV A Case Study in Robot Exploration 59

V Perception for Rugged Terrain 74

Abstract

This report describes progress in research on an autonomous robot for planetary exploration performed during 1988 at the Robotics Institute, Carnegie Mellon University. This report begins with an introduction, summary of achievements, and lists of personnel and publications. It also includes several papers resulting from this research.

This research is primarily sponsored by the National Aeronautics and Space Administration. Portions of this research are also supported by the National Science Foundation and the Defense Advanced Research Projects Agency.

The research program includes a broad agenda in the development of an autonomous mobile robot. In the year covered by this report, we addressed four major topics:

Configuration: We configured the *Ambler* (acronym for Autonomous **MoBiLe** Exploration **R**obot) as a walking robot to overcome the shortcomings exhibited by existing wheeled and walking robot mechanisms. The fundamental advantage of the Ambler configuration—which has implications for efficiency, mechanism modeling, and control simplicity—is that actuators for body support are independent of those for propulsion; a subset of the planar joints propel the body, and the vertical actuators support and level the body over terrain.

Perception: To characterize local scene geometry, we developed several different techniques for constructing elevation maps from range images provided by a scanning laser range finder. Further, we developed techniques to (1) evaluate elevation map regions as footfall locations, (2) match elevation maps, and (3) extract topographic terrain features from elevation maps (peak, pit, ridge, valley, etc).

Planning: We developed innovative gaits that exploit the novel ability of the Ambler to recover a trailing leg past a leading leg to achieve productive locomotion and flexibility in selection of footfalls.

System Integration: We developed and partially implemented a centralized task control architecture to integrate the perception, planning, and control algorithms. We experimented with integrated systems on two separate testbeds. For the first testbed, we designed and fabricated one full-scale leg of the proposed walking vehicle. We demonstrated single-leg motion with simplified perception, planning, and control modules. Further leg testing has provided results on mechanical and structural integrity, leg recovery planning, foot slippage, and power consumption. The second testbed is a simple wheeled mobile manipulator. We developed algorithms for combining perception, locomotion, and manipulation to control the robot in an indoor laboratory to collect simple objects. This work allows us to research complex planning and control issues in parallel with developing the Ambler mechanism.

Introduction

This report reviews progress during 1988 at the Robotics Institute, Carnegie Mellon University, on research sponsored by NASA titled "Autonomous Planetary Rover." This report begins with an overview and a summary of achievements. It then lists the members of the research group supported by, or directly related to the contract, and their publications. Finally, it includes four detailed papers representative of specific areas of research.

Overview

The CMU program to develop an Earth-based prototype of an autonomous planetary rover is organized around three teams that are developing the locomotion, perception, and planning subsystems (figure 1). A joint task is to integrate the three subsystems into an experimental robot system. We will use this system for evaluating, demonstrating, and validating the concepts and technologies developed in the program.

The technical objectives of the research include the following:

- To develop and demonstrate an autonomous Earth-based mobile robot that can survive, explore, and sample in rugged, natural terrains analogous to those of Mars.
- To provide detailed, local representations and broad, 3-D descriptions of rugged, unknown terrain by exploiting diverse sensors and data sources.
- To demonstrate robot autonomy through a planning and task control architecture that incorporates robot goals, intentions, actions, exceptions, and safeguards.

One of the major accomplishments of 1988 is the configuration of the legged vehicle sketched in figure 2. This unprecedented walking robot provides locomotion over rugged terrain that wheeled vehicles can not negotiate easily, and promises to be simpler to control and more reliable than other walkers without orthogonal legs. Another major accomplishment of 1988 is the construction and utilization of the experimental system hardware shown in figure 3. This full-scale testbed not only accelerates the development and testing of perception, planning, and control algorithms, but also focuses the integration efforts required to create a coherent robotic system from component research results.

Our overall objectives for the coming year are (1) to develop the hardware for a multi-legged walking robot, (2) to continue integrating the locomotion, perception, and planning subsystems, and (3) to continue developing the technological basis required for a fully integrated robot system.

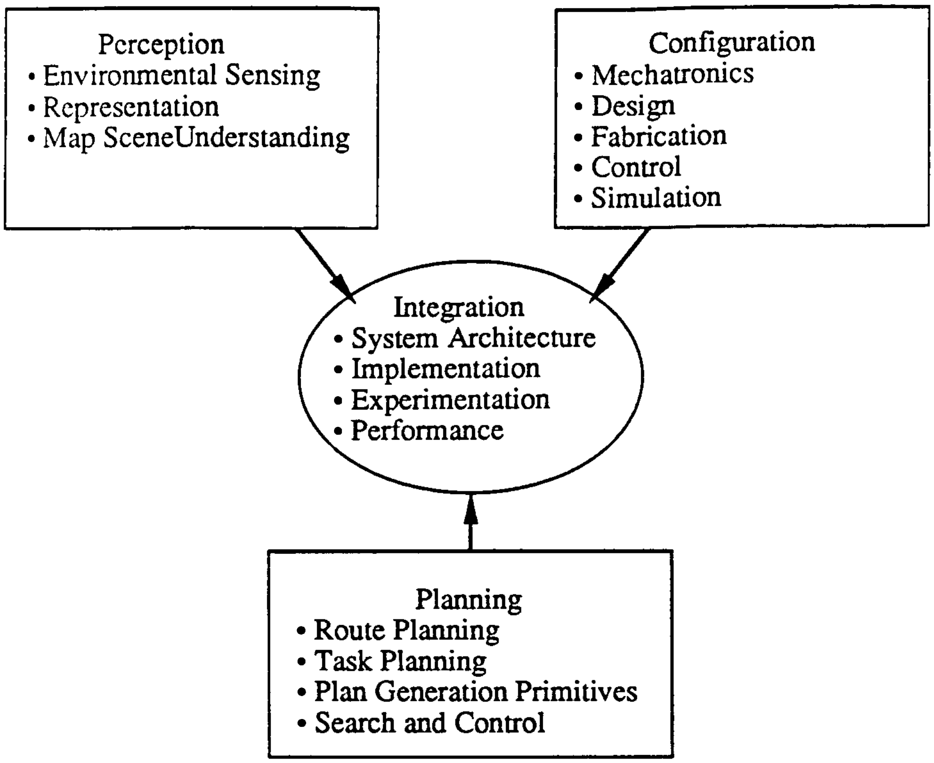

Perception
• Environmental Sensing
• Representation
• Map SceneUnderstanding
Configuration
• Mechatronics
• Design
• Fabrication
• Control
• Simulation
Integration
• System Architecture
• Implementation
• Experimentation
• Performance
Planning
• Route Planning
• Task Planning
• Plan Generation Primitives
• Search and Control

Figure 1: Research program organization

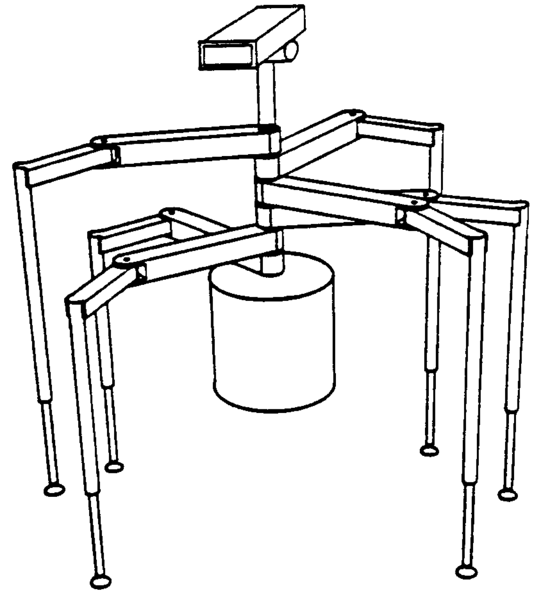


Figure 2: The Ambler

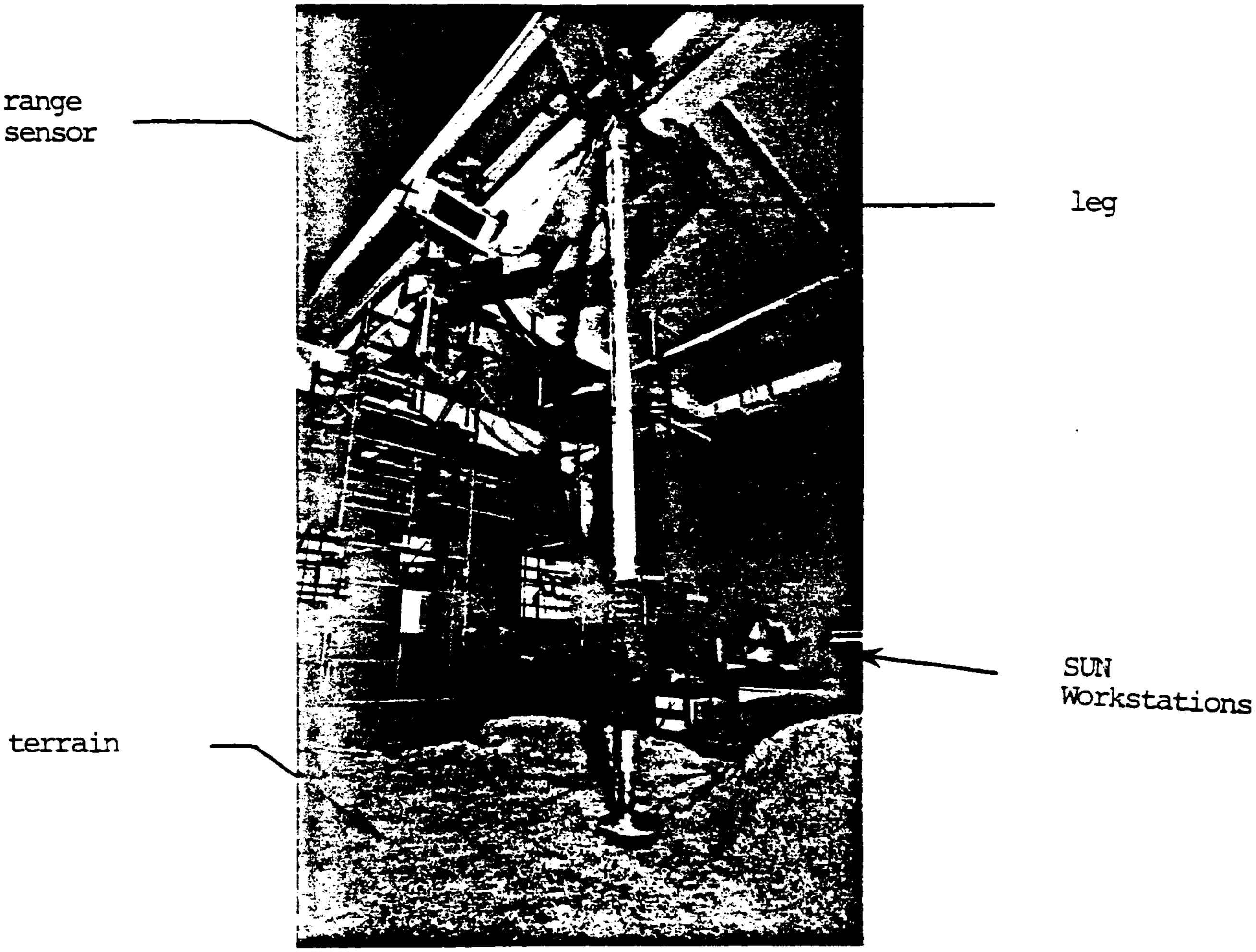

range sensor
leg
SUN Workstations
terrain

Figure 3: Single leg testbed

Accomplishments

This section describes the key accomplishments of the research in the time period from January 1988 to December 1988.

Configuration We configured the Ambler (acronym for Autonomous **MoBiLe** Exploration Robot) as a walking robot to overcome the shortcomings exhibited by existing wheeled and walking robot mechanisms. Walking is especially suited for autonomous locomotion as efficiency is theoretically high, motions are predictable, incremental, and assessable, and motions can be used to position and orient body-mounted sampling equipment. We converged on legged locomotion since wheeled and tracked mechanisms exhibited limited traversability and often unpredictable body motions. We rejected hybrid locomotion and manipulator-assisted mechanisms (e.g., "strong arms") because of planning and control complexity, and the significant cost of weight, scale, and power required for the additional mechanisms.

The fundamental advantage of the Ambler configuration—which has implications for efficiency, mechanism modeling, and control simplicity—is that actuators for body support are independent of those for propulsion; a subset of the planar joints propel the body, and the vertical actuators support and level the body over terrain. The Ambler is capable of overlapping gaits where rear legs recover *past* forward supporting legs. These overlapping gaits mean fewer foot placements with advantages of reduced demand on perception and planning, and significant energy savings due to reduction of the number of foot-terrain interactions. Any functional leg can reposition itself to substitute for any failed leg, thus significantly enhancing mechanism redundancy.

Mechanism Model We formulated a functional model of legged locomotion on natural terrain for use in model-based control schemes, simulation studies, and design evaluations. The model considers closed chains, non-conservative compliance at the foot-soil interface, full non-linear dynamic response, and solution techniques for both forward and inverse dynamic calculations. The forward dynamics are solvable as singular systems of differential equations. Substitution of difference expressions convert the inverse dynamics to a form that is solved by Newton's method.

Perception using Elevation Maps To characterize local scene geometry, we developed several different techniques for transforming range images into elevation maps. The range images are acquired by a scanning laser range finder developed by the Environmental Institute of Michigan. Simulations showed one of them—the Locus Method—to be extremely robust to sensor noise and least sensitive to surface orientation. We demonstrated the method with ERIM images of rough terrain acquired at a construction site; the resulting elevation maps accurately capture the scene. We also developed techniques for smoothing and interpolating elevation maps, and evaluating elevation map regions as footfall locations.

Perception of Terrain Features We identified topographic terrain features (peaks, pits, ridges, valleys, etc) to extract from elevation maps for the purpose of terrain typing. These features appear to be more relevant for sampling operations and less relevant for navigation and locomotion.

Walk Planning We developed innovative gaits that exploit the novel ability of our proposed walking vehicle to recover a trailing leg past a leading leg to achieve productive locomotion and flexibility in selection of footfalls. We developed a criterion that ensures stability of a six-legged walker, even in the event of spontaneous failure of any leg. The resulting gaits provide the benefits of periodicity in that they exhibit regular sequence and timing, but allow foot placements over generous regions, which is essential in irregular terrain. In addition, they should require fewer footfalls per unit advance, thus improving efficiency.

Architectures We completed initial studies of several possible planning and control architectures, including those with explicit goals for both the world state and processing, multiple controllers with varying competences plus a mediation mechanism, "reactive" systems, "subsumption" systems, dynamic hierarchical systems, and blackboard systems. We converged on a centralized task control architecture, and have implemented a preliminary version.

Single Leg Testbed We designed and fabricated a full-scale single-leg testbed. It incorporates an actuator to simulate body motion so that propulsion and leg recovery can be accurately studied. It includes different types of terrain so that foot contacts and slippage can be thoroughly evaluated. A range sensor mounted above the leg is used to build terrain maps and select footfall areas. The leg testing program has provided results on mechanical and structural integrity, leg recovery planning, foot slippage, and power consumption.

Mobile Manipulator Testbed This testbed is based on a modified commercially available mobile robot with arm. We have configured it with a video camera mounted in the ceiling of lab and three sonar sensors on the robot (one fixed, one rotating, one on wrist of gripper). This testbed supports development of our task control architecture, and future experiments with error detection and recovery, planning of sensor positioning, and studies of large-delay teleoperation.

We developed computer programs enabling the mobile manipulator to recognize, locate, navigate to, plan paths to, and manipulate a small set of fixed objects in the mobile manipulator testbed. We developed these programs to obtain experience with issues such as integrating perception, planning, plan execution, and error detection. They have also raised issues such as interactions between locomotion and manipulation planning, and determining which sensors and data resolutions are most useful for various subtasks.

System Integration We incorporated preliminary integration efforts into configuration and design of the proposed robot, as decisions were made about cooperation among electromechanisms, sensing, modeling, planning, control, and human interaction. We demonstrated single-leg motion with perception, planning, and control modules coordinated by the task control architecture.

Personnel

Directly supported by the project, or doing related and contributing research:

Faculty: Martial Hebert, Takeo Kanade, Eric Krotkov, Tom Mitchell, Reid Simmons, Chuck Thorpe, William Whittaker.

Staff: Purushothaman Balakumar, Mike Blackwell, Kevin Dowling, Christopher Fedor, Regis Hoffman, Ralph Hyre, Chelva Kumar, Clark MacDonald, Jim Moody, Henning Pangels.

Visiting Scientists: Claude Caillas.

Graduate Students: John Bares, Lonnie Chrisman, Joe Hirsch, In So Kweon, Long-Ji Lin, Swami Mahalingam, Dave Manko, Mike Murawski, Peter Nagy, Marc Ringuette, Dave Wettergreen.

Undergraduate Students: Chris Ivory, Nina Koros, Andy Phillips, Bruce Thompson, Rob Yohe, Kurt Zimmerman.

References

[1] J. Bares. Orthogonal Legged Walkers for Autonomous Navigation of Rugged Terrain. December 1988. Ph.D. Thesis Proposal, Department of Civil Engineering, Carnegie Mellon University.

[2] J. Bares and W. Whittaker. Configuration of an Autonomous Robot for Mars Exploration. In *Proc. World Robotics Conference*, Society of Mechanical Engineers, Gaithersburg, Maryland, To appear, May 1989.

[3] Planetary Rover Group. Experiments in Perception, Planning, and Control for the Carnegie Mellon Mars Rover. December 1988. Robotics Institute, Carnegie Mellon University, unpublished working document.

[4] Planetary Rover Group. Integration of Perception, Planning, and Control in the Carnegie Mellon Mars Rover. December 1988. Robotics Institute, Carnegie Mellon University, unpublished working document.

[5] M. Hebert, T. Kanade, E. Krotkov, and I. S. Kweon. Terrain Mapping for a Roving Planetary Explorer. In *Proceedings of IEEE Robotics and Automation Conference*, Scottsdale, Arizona, To appear, May 1989.

[6] E. Krotkov, J. Bares, M. Hebert, T. Kanade, T. Mitchell, R. Simmons, and W. Whittaker. An Autonomous Rover for Exploring Mars. *Robotics Institute Annual Review, Carnegie Mellon University*, To appear, March 1989.

[7] E. Krotkov, J. Bares, M. Hebert, T. Kanade, T. Mitchell, R. Simmons, and W. Whittaker. An Autonomous Rover for Exploring Mars. *IEEE Computer*, To appear, June 1989.

[8] E. Krotkov, C. Caillas, M. Hebert, I. S. Kweon, and T. Kanade. First Results in Terrain Mapping for a Roving Planetary Explorer. In *Proc. NASA Conference on Space Telerobotics*, Jet Propulsion Laboratory, Pasadena, California, January 1989.

[9] I. S. Kweon, M. Hebert, and T. Kanade. Perception of Rugged Terrain. In *Proc. SPIE Mobile Robots III Conference*, Society of Photo-Optical Instrumentation Engineers, Cambridge, Massachusetts, November 1988.

[10] I. S. Kweon, M. Hebert, and T. Kanade. Sensor Fusion of Range and Reflectance Data for Outdoor Scene Analysis. In *Proc. NASA Workshop on Space Operations, Automation, and Robotics, NASA Conference Publication 3019*, pages 373–382, Dayton, Ohio, July 1988.

[11] L. J. Lin, T. M. Mitchell, A. Phillips, and R. Simmons. *A Case Study in Robot Exploration*. Technical Report CMU-RI-TR-89-1, Robotics Institute, Carnegie Mellon University, January 1989.

[12] S. Mahalingam. *Terrain Adaptive Gaits for the Ambler*. Master's thesis, Department of Mechanical Engineering, University of North Carolina, 1988.

[13] D. Manko. Models of Legged Locomotion on Natural Terrain. June 1988. Ph.D. Thesis Proposal, Department of Civil Engineering, Carnegie Mellon University.

[14] D. Manko and W. Whittaker. Inverse Dynamic Models used for Force Control of Compliant Closed-Chain Mechanisms. In *Proceedings of IEEE Robotics and Automation Conference*, Scottsdale, Arizona, To appear, May 1989.

[15] P. Nagy. Attitude and Altitude Control for a Novel 6-Legged Robot. January 1989. Research Prospectus, Mechanical Engineering Department, Carnegie Mellon University.

[16] P. Nagy and D. Manko. Preliminary Results on Foot/Terrain Interaction for a Walking Robot. In *Proceedings of Robots 13 Conference*, Gaithersburg, Maryland, To appear, May 1989.

[17] P. Nagy and W. Whittaker. Motion Control for a Novel Legged Robot. In *Proceedings of IEEE Robotics and Automation Conference*, Scottsdale, Arizona, To appear, May 1989.

[18] W. Whittaker and J. Bares. Orthogonal Legged Walking Robot. December 1988. United States Patent Application.

An Autonomous Rover for Exploring Mars

John Bares Martial Hebert Takeo Kanade
Eric Krotkov Tom Mitchell Reid Simmons William Whittaker

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We present an overview of a comprehensive research program, begun in 1987, to design and build an autonomous intelligent machine to undertake an exploratory prospecting mission on Mars. We are not attempting to satisfy all constraints on the system that would be flown to Mars; we are building a prototype and testing it on full-scale Mars-like terrain. The research focuses on the central robotics issues of locomotion, perception, planning, control, and sample manipulation; the article describes how we confront the issues not faced by laboratory robots and explains our system designs. An unprecedented walking robot provides locomotion over terrain that wheeled vehicles cannot negotiate easily; its six legs are stacked coaxially, permitting each to rotate fully about the body. The perception system constructs and matches multiple-resolution maps of rugged terrain and discrete objects using diverse sensors; we present a detailed example of how it builds elevation maps from a sequence of laser rangefinder views. A centralized planning and control architecture smoothly integrates different performance level planners and provides for flexibility in the detection and handling of unexpected contingencies. We present a preliminary discussion of sampling tasks and requirements; we propose simple, task-specific tools to acquire samples of Martian materials by cutting, coring, scooping, and raking.

Notice: To appear in Special Issue of Computer Magazine on Autonomous Intelligent Machines, June 1989.

Contents

1 Introduction **11**

2 Exploring Mars **12**

3 Locomotion **13**

4 Perception **15**
4.1 Representations 16
4.2 Constructing Terrain Maps 17
4.3 Using Terrain Maps 20

5 Planning and Control **20**
5.1 Integration 22
5.2 Flexibility 23
5.3 Self-Awareness 25

6 Sampling **26**

7 Current Status **28**

8 Summary **29**

Acknowledgements **30**

References **31**

Figures **32**

1 Introduction

For centuries, people have been fascinated by Mars, one of our nearest planetary neighbors. There has been much speculation, in science and science fiction, about what lies on and under the surface of the Red Planet. Despite considerable study, our knowledge remains very limited. Orbiting vehicles cannot examine internal features, and stationary vehicles, like the three Soviet and two US landers, miss what is over the horizon, atop mountains, and in ravines.

Active exploration of Mars could answer many questions about the nature and origins of our solar system. Sending astronauts or remotely controlled vehicles are possibilities, but a manned Mars expedition is highly unlikely within the near future, and conventional teleoperation is impractical for such a mission because of the long signal time to Mars (up to 45 minutes for a round trip at the speed of light). A more promising approach is NASA's current plan to launch a "Mars rover and sample return" mission in 1996, involving an unmanned prospector and a vehicle to return collected samples to Earth [6]. The broad objectives in exploring Mars are to observe and gather materials representative of the planet's geophysical, meteorological, and biological conditions, and to return a variegated selection of samples. The mission could last two years, during which the rover might traverse hundreds of kilometers. Since the payload of the return vehicle is limited, the mission requires a sophisticated on-site system that can explore, assay, evaluate, and select.

In October 1987 we initiated a research program that addresses the core robotics challenges of designing a roving explorer capable of operating with minimal external guidance. This research is needed to confront issues not faced by laboratory robots, to identify and formulate the difficult problems in autonomous exploration, and to generate the insights, principles, and techniques for their solution. We are not attempting to satisfy all constraints on the system that would be flown to Mars (e.g., space-qualified processors). Instead, we are building a prototype legged rover (called the *Ambler*[1]) and testing it on full-scale, Mars-like terrain.

To undertake a prospecting mission on the Martian surface, we must extend existing robotic technology. Because a Mars-roving system is beyond the reach of timely aid from Earth, it must exhibit extreme self-reliance. The rover must be able to navigate, explore, and sample within its abilities, and to know, moreover, what tasks do and do not lie within its capabilities. Particular issues critical to autonomous planetary exploration include robust rough terrain navigation, capable locomotion, sample acquisition, perception, self-awareness, task autonomy, safeguarding, and system integration. While semi-autonomous and tele-assisted systems may be practical for some tasks [11], our research strategy is to strive for full autonomy wherever possible, and to have the rover decide when to ask for missing information.

In this article, we present an overview of our research program, focusing on the core areas of locomotion, perception, planning, and sampling. Since the program is less than one year old, this article aims to identify issues and approaches and to describe work in progress, rather than to report results. While our exposition concentrates on a Mars mission, we expect many of the technologies to be applicable both to other planetary bodies and to terrestrial concerns such as hazardous waste assessment and remediation, ocean floor exploration, and mining.

---

[1] Ambler is an acronym for Autonomous MoBiLe Exploration Robot.

2 Exploring Mars

Our current data on the Martian landscape indicates that an explorer would encounter a wide variety of terrain features, including a canyon 4800 km long by 7 km deep, a mountain 27 km high, and numerous sand dunes, rock fields, and craters. For example, figure 1 illustrates the barren, rugged terrain viewed by the Viking 2 lander.

The following hypothetical scenario illustrates some of the challenges the rover must meet in actively exploring such terrain: The rover is instructed to obtain core samples from a rock outcrop several kilometers away. Using an area map at 10-meter resolution provided by an orbiter, the rover plans a path to the rock face which skirts a small crater that lies on the direct route. As it traverses the path, the rover uses cameras and rangefinders to survey the immediate region, building a geometric map that reveals a large boulder field like that in figure 1. The rover uses this local map to select where to place its feet and how to move its legs in order to walk safely through the meter-high boulders.

A team of Earth-based scientists monitors the rover's progress in a time frame that lags behind the action. In data transmitted as the rover approached the outcrop, they notice a vertical formation apparently containing several strata. The team deems it desirable to obtain a sample from each layer, and relays this information to the rover, which charts a course to the indicated stratum.

Close to the rock wall the rover encounters a sandy, rocky incline. The rover judges the steepness to be well within its capabilities, and proceeds to climb up the slope. As feet sink in sandy areas, attitude control systems reflexively alter leg length to maintain a level body and ensure stability. Noting the surface's visible features, the rover alters its planned trajectory to avoid similar areas of potentially treacherous footing.

Once in position, the perception system constructs a high-resolution model that reveals four distinctly colored layers. The rover then chooses points centered in each band, orients itself, deploys a core drill to extract core sections, and stores them for delivery to the return vehicle.

This scenario illustrates many of the issues important for a planetary explorer — autonomous locomotion and sampling, navigation over a wide range of terrains, multi-sensory perception, perception at multiple levels of resolution, both long-range planning and reactive planning for contingencies, and awareness of the rover's own capabilities and limitations. It also points to some of the goals for and constraints on a rover design. In the next sections, we summarize some of the most important constraints, and describe how locomotion, perception, planning, and sampling systems can satisfy these constraints.

3 Locomotion

The rover locomotion system must safely transport the vehicle over vast expanses of the irregular Martian terrain. Perhaps the most important design criteria for the locomotion system is traversability: it must be able to navigate over extremely rugged terrain. Specifically, it is desirable that the rover be capable of traversing a one meter step, negotiating a 60 per cent slope, and maintaining an average velocity of approximately 1 km/day[2]. Autonomous operation places additional traversability constraints, namely, that the locomotor be capable of incremental three-dimensional motions that are predictable and reversible. The ability to move incrementally in any direction greatly simplifies path planning in rugged terrain since the mechanism places few constraints on the planners. Predictable motion, knowing that the locomotor will move as commanded, is the basis for safe and reliable traversal of difficult terrain. Finally, the ability to reverse any motion sequence at any time is important to all vehicles that venture into the unknown.

Energy efficiency poses an additional design constraint, because total on-board power generation is expected to be less than 1 kw. As the dominant energy consumer the locomotion mechanism must be extremely efficient. Another design consideration is that the locomotion mechanism must provide a stable platform both for sensors and sample acquisition tools.

These design criteria admit a wide variety of possible locomotion candidates, including mechanisms that roll, walk, combine rolling and walking, or perform so-called hybrid locomotion [12]. Rolling machines have wheels (or tracks) in continuous support contact with the terrain and propel themselves by generating traction forces parallel to the terrain surface. They are the predominant form of locomotion for most manned vehicles and unmanned robots. As a result, their control and performance trade-offs are well-understood. Walkers suspend themselves over the terrain on discrete contact points and maintain principally vertical contact forces throughout propulsion; this allows more tractable models of terrain interaction than are possible for wheels. In addition, walking mechanisms isolate the robot's body from the underlying terrain and can propel the body along a smooth trajectory independent of surface irregularities.

After comparing these candidates, we selected legged locomotion because of its superior rough terrain traversability characteristics, theoretical efficiency, and its ability to keep sensors and sampling equipment steady and stable. A thorough trade-off analysis of locomotion mechanisms with respect to these constraints appears in [1].

The Ambler design [1] consists of six legs stacked coaxially at their shoulder joints (see figures 2,3). Each leg is mounted at a different elevation on the central axis of the body and can rotate fully around the body. Each leg (figure 4) consists of two revolute joints (shoulder and elbow) that move in a horizontal plane to position the leg, and a prismatic joint at the end of the elbow link that effects a vertical "telescoping" motion to extend or retract the foot. Thus, the locomotor has 18 degrees of freedom. The planar "reach" (combined length of shoulder and elbow links) of a leg is 2.5 m and vertical "stroke" (telescopic distance) is 1 m. The average overall height of the Ambler is approximately 3.5 m, and its nominal width is approximately 3 m. With these dimensions, the Ambler can step over obstacles 1 m high while maintaining a level body trajectory — we know of no other robot that can perform nearly as well.

---

[2] These specifications, although somewhat arbitrary, reflect plausible assumptions about the mission and the scale of objects on the Martian surface.

The Ambler body, a 1 m diameter cylinder situated below the leg stack, will contain equipment for power generation, computing, sample analysis, and scientific instrumentation. Sample acquisition tools may be mounted beside or under the body (see section 6, ff.). Communication equipment (not shown in figure 3) can be mounted either above the leg stack or in the body. Perception sensors, which currently include a laser rangefinder and color cameras, will be mounted above the leg stack where they have larger fields of view; other high-resolution sensors may be placed under the body or directly on the legs.

This configuration possesses a number of benefits. First, the long legs isolate the body and sensors from terrain roughness. This is an advantage for perception; since the body remains level and at the same altitude, sensor observations are aligned with each other over time much more than they would be on a rolling vehicle. Second, decoupling the vertical and horizontal joints simplifies walk planning and motion control by reducing complex six-dimensional problems to smaller ones. Third, sampling tools under the body have a clear view of and close proximity to the terrain that they must access, and can be positioned and oriented by moving the body, reducing the number of degrees of freedom required for sampling equipment.

In operation, the Ambler will walk over rugged terrain much as one poles a raft floating on water over a rough lake bottom. The six vertical actuators in the Ambler's legs level the body over terrain, while the planar joints propel the body. As the body advances, one leg at a time moves ahead of the walker, much like the pole is replaced ahead of the raft. A unique result of the stacked leg configuration is that gaits where rear legs recover past forward supporting legs are possible. Figure 5 shows an example of such an *overlapping* gait. Overlapping gaits require fewer foot placements, saving energy because of fewer terrain deformations, and reducing demands on perception and planning.

While recovering one leg, the five other legs support the body. The stability of the stance can be maximized by maintaining the center of gravity inside a "conservative support polygon." Inside this region, the vehicle remains stable even if one (and possibly more) of the legs ceases to support the vehicle, either due to failure or slippage.

Experience with existing walking mechanisms [5] suggests that they are difficult to coordinate due to their complexity, suffer large energy losses due to actuator conflict, and can be unreliable upon failure of one or more legs. We designed the Ambler to overcome each of these three problems.

Unlike other walkers, the Ambler's actuator groups for body support and propulsion are orthogonal; a subset of the planar joints propel the body, while the vertical actuators support and level the body. The Ambler can level itself without propelling and propel without leveling, exhibiting no power coupling between the two motions. A subset of the planar joints is sufficient to propel the body in position (x, y) and heading θ, because the planar mechanism (12 links) is a determinate system. Any three of the twelve shoulder and elbow actuators can be used to propel the body, with the remaining nine planar actuators declutched or back-driven[3]. Since the planar mechanism is determinate (three actuators provide three degrees of body freedom), actuator conflict and the ensuing energy losses are eliminated. Planar dynamic response can also be completely modeled for use in motion control of the mechanism. A one-dimensional terrain interaction model suffices for solution of the vertical foot forces. Equilibrium, stability margin, forces, and energy cost are readily computable from these mechanism and terrain models, which among other benefits, permits torque and power minimization during propulsion.

The Ambler locomotor configuration is a dramatic improvement in reliability over conventional walking mechanisms. Since the legs are stacked above the body and can rotate by 2π about their shoulder joints, any leg can operate in any body sector. Thus, any functional leg can reposition itself to substitute for any failed leg, and three legs would have to fail to cause immobilization.

In summary, the Ambler is an unprecedented walking mechanism that satisfies the constraints of traversability, efficiency, and stability imposed by the rugged Martian surface and mission requirements. Its unique design avoids problems faced by other walkers, and qualifies it for the aggressive exploration task it faces.

4 Perception

The Ambler needs timely and detailed perception to plan effective locomotive and sampling strategies, monitor their execution, and safeguard against hazards such as tipover or environmental change. These capabilities require appropriate and efficient data representations for terrain (e.g., geometry, soil type) and objects (e.g., size, shape). The perception system's task is to build and maintain these representations — of the terrain, its properties, and discrete objects — which we call *terrain maps.*

While orbiters and Earth-based controllers can assist the mission, they cannot provide the perceptual basis for navigation or sampling tasks. Maps produced by orbiters will not have sufficient resolution for sampling or mobility in very rough terrain — even the highest resolutions currently envisioned will yield maps with only meter-scale resolution. Due to communication delay, Earth-based interpretation of sensor data would drastically limit the rover's speed even over the most benign terrain. Thus, the Ambler itself must have the sensing and perception capabilities to build the detailed terrain maps and object descriptions needed for local motion and sampling.

---

[3] However, combinations of three elbow actuators cannot provide body heading. Furthermore, mechanism singularities exist for which certain shoulder and elbow actuators require infinite torques.

Perceiving and mapping rugged, outdoor terrain poses significant challenges. Current machine perception techniques can be applied with some success to man-made, structured, indoor scenes. Unlike industrial systems, the rover will have little need to recognize or describe regular geometric shapes, and cannot capitalize on the powerful constraints (e.g., symmetry, smoothness, constant illumination) presently utilized to perceive worlds consisting of blocks and origami pieces. New techniques must be developed to construct maps of the natural, unstructured, outdoor environment of Mars.

Building and maintaining those maps raises several issues: 1) representation of data at different levels of resolution; 2) construction of maps and descriptions from different sensors; and 3) efficient use of the maps. The following three sections describe how the perception system addresses these issues.

4.1 Representations

The perception system must provide an environmental representation that is appropriate for a wide variety of tasks, each with different requirements. For example, locomotion and sampling require detailed, local representations, while navigation and mission planning demand broad, global descriptions. To accommodate these diverse needs in a uniform fashion we have selected a hierarchical representation scheme which describes terrain and objects at varying levels of resolution.

According to the needs of different tasks, it is natural (but not necessary) that the resolutions differ by orders of magnitude: pebble-size units on the order of 1 cm may be appropriate for performing fine sampling operations; foot-size units on the order of 10 cm might be well-suited for locomotion control; vehicle-size units on the order of 1 m may be natural for navigation and short-range path planning. Figure 7 illustrates this idea for three specific levels of resolution. The first two rows of the figure suggest how a map (called the global map because it is created from orbiter images) at 10 m scale could be used to plan a 1 km route (perhaps a day's journey). The third row of the figure shows how the global map relates to a local map at 1 m scale. The fourth row suggests first how the 1 m scale map could be used to plan a 10 m path, and then how a 10 cm scale map (see inset) could be used to select footfall locations within a 1 m region.

In addition to making available information at appropriate scales, the multi-resolution representation affords several computational advantages. For functions applicable only at certain resolutions it reduces the amount of data that must be accessed. Functions applicable at all resolutions may be evaluated more efficiently by processing all resolution levels in a coarse to fine manner, or by processing only those levels containing useful information.

At each level of resolution we describe the environment in two ways: on a geometric grid, and as object descriptions. Together these comprise a terrain map at one scale.

We define an *elevation map* on a regular grid. Each grid square records information about the terrain in that area, for instance, its elevation above a reference "ground" plane (see figure 8). Other terrain attributes include the following: the uncertainty of the estimated elevation; roughness; slope; labels indicating whether the terrain is unknown (has never been observed) or occluded (currently not observed because it lies in a shadow cast by another object); mineralogical composition; and a measure of traversability derived from slope, roughness, and other properties.

Object descriptions include the size, shape, and location of particular objects such as a boulder; symbolic descriptions of terrain such as hill, valley, saddle, and ridge that may be useful for identifying promising sample sites; paths the vehicle has followed; locations that have been sampled; and viewpoints from which observations have been made.

4.2 Constructing Terrain Maps

Constructing terrain maps requires sensing and interpretation, ranging from low-level data collection to high-level scene modeling. This section first focuses on the lowest level of abstraction: sensors and signals. Next it describes an intermediate level of abstraction, concentrating on local surface geometry and local material properties, and then sketches the highest level of objects and semantic interpretations.

Sensors

Single-sensor, single-algorithm systems are severely limited in their ability to resolve ambiguities, to identify spurious information, and to detect errors or failure. These shortcomings are not a product of the sensors or algorithms employed, they are an unavoidable consequence of attempting to make global decisions based on incomplete and underconstrained information. One way to circumvent the limitations of a single sensor is to use multiple sensors. By combining ("fusing") information from many different sources, it is possible to reduce the uncertainty and ambiguity inherent in making decisions based on only a single information source. We will equip the Ambler with a battery of different sensors (even using the vehicle itself as a sensor) to collect multi-spectral data and to allow sensor fusion.

Currently, our primary sensor is a scanning laser rangefinder that measures both reflectance and range. It has two disadvantages: its moving parts are vulnerable to failure; and since the scanner actively transmits signals, it requires more power than do passive imaging systems. However, we find that its virtues — directly measuring the environment's three-dimensional structure — easily outweigh its defects; compared to reconstruction of depth from multiple two-dimensional images (e.g., stereo, motion), it imposes a far smaller computational burden, and eliminates the errors that inevitably arise in solving the correspondence problem. We have developed an explicit probabilistic model of the uncertainty on this sensor's readings, according to which the measured range errors are normally distributed with standard deviation proportional to the square of measured range.

In the near future we will also use a pair of color cameras in order to determine material properties from color and texture, for long-range viewing, and for stereo viewing to back up the rangefinder and to aid mission controllers on Earth. In later stages of our research we plan to incorporate proximity sensors that detect imminent collisions with obstacles, an inertial reference sensor, inclinometers, tactile sensors on sampling tools, and other imaging devices.

Interpreting the raw sensor measurements cannot begin until the devices are calibrated and their signals corrected. Calibration involves relating the sensor measurements to a known quantity (e.g., the dimensions of a leg or the reflectance of a reference pattern) or to each other (e.g., the geometric relationship of cameras to rangefinder). Periodically, using well-known techniques [8], the rover must recalibrate the devices as they bounce and shake out of alignment. Correction involves detecting and eliminating corrupt signals (e.g., images acquired with the camera pointed at the sun), and filtering noisy signals.

Surface geometry

Once calibrated and corrected, the perception system can interpret sensor signals at higher levels of abstraction. To give an example of this important operation, we will consider the interpretation of rangefinder signals in terms of local surface geometry (interested readers can find details in [3]). This involves creating an elevation map from a range image, computing its geometric attributes, matching it to another elevation map, and merging the two maps to form a composite map.

A simple method to create the elevation map is to refer each range measurement to a reference grid representing the ground plane by applying a coordinate transformation determined during calibration. This approach is limited since it cannot compute the elevation estimate of a point that is not a grid point without resampling. We have developed an algorithm overcomes this difficulty by first computing *in the image space* the locations where rays transmitted by the sensor strike the terrain, and then referring the intersection points and an estimate of their uncertainty to a reference grid of arbitrary resolution.

We have implemented algorithms to compute geometric attributes of the surface defined by the elevation map, including surface normal, principal surface curvatures, shadowed areas, and elevation uncertainty. Slope and roughness can be calculated by identifying the best planar fit to the surface: slope can be determined from the plane's parameters; roughness can be derived by examining the fit residual. We are investigating techniques to estimate surface roughness that compute the fractal dimension of the set of surface points.

So far, the characterization of surface geometry derives from a single range image acquired at one fixed location. Merging elevation maps from successive viewpoints allows the construction of a composite map that can be more complete, by including previously occluded regions, and more accurate, by decreasing the uncertainty of past measurements. Further, determining the geometric relationship between the viewpoints provides a measure of the vehicle displacement which can be used to supplement position measurements from dead reckoning and inertial references. We have developed a two-stage algorithm to determine the correspondence between two elevation maps. The first stage matches a sparse set of geometric features extracted from the two maps using hypothesis prediction and verification; its output is the estimated rigid transformation T relating the two sets of features. The second stage takes T as an initial estimate and refines it by gradient descent, iteratively minimizing an error functional defined over all the data points in the two maps. Once we know T, we can apply it to merge maps; figure 9 illustrates an example of a composite elevation map constructed by merging four rangefinder views of the rugged terrain at a construction site. In addition to computing composite elevation maps, we can apply the algorithm to the problem of registering local maps with global maps (figure 7).

Material properties

The mobility of the Ambler will be determined not only by the geometry of surfaces but also by their material properties. To characterize a surface as sand, rock, volcanic ash, or talus is a formidable task, requiring a variety of information sources. Although we have not yet formulated solutions to this problem, we have identified three approaches. First, the perception system can apply statistical pattern recognition and sensor fusion techniques to classify materials based on complementary information such as color, visual texture, and surface roughness measures derived from range and color images. Second, it might apply reasoning techniques to infer a lower bound for particle cohesion from a measured surface's maximum slope and some geological knowledge. Third, the perception system need not be limited to passively interpreting data; it can *actively* use the Ambler vehicle itself as a sensor to determine soil cohesion and friction parameters either directly, by measuring leg joint torques while walking, or indirectly, by comparing the soil in its footprints to nearby soil.

Object identification

Interpreting sensor signals at the highest level of abstraction, in terms of objects and their semantics, is a very important capability. For example, ices condensing during the cold Martian night may evaporate before scientists on Earth can notice them and direct the rover to collect them — they must be discovered and analyzed *in situ*. To do this, we hope to build special-purpose object identification routines. To extend these to the point where they can 1) identify "unusual" or "interesting" objects, perhaps on the basis of color, texture, or context, and 2) provide the basis for autonomous decisions to collect a sample, is well beyond the state of the art. Studying these problems of semantic object identification have high priority on our future research agenda.

4.3 Using Terrain Maps

Once maps and descriptions of the world have been constructed, the perception system must support and facilitate their use. It will provide multiple access mechanisms to its data representations, including at least the following: efficiently answering a single query, perhaps by evaluating a spatial hash function; receiving an interrupt, for instance a request for particular data; and continuous output streaming, to be used in actuator control loops.

The terrain maps will be used for a variety of tasks. For locomotion, the Ambler will access elevation maps to select footfall locations that can both accommodate its feet and support its mass. It will also compute intersections between elevation maps and planned trajectories to ensure that its body and legs avoid obstacles. For navigation, the Ambler will use the elevation maps to plan paths and routes. Further, it will use the elevation maps to localize itself by matching them to global maps, and combine this information with position estimates from dead reckoning and inertial references. For sample acquisition, the Ambler will use both elevation maps and object descriptions: the former to identify promising sampling sites based on topographic features; the latter to identify objects to be sampled, determine approach directions, and select control regimes (e.g., force, position).

To summarize, the perception system provides detailed, local representations and broad, three-dimensional descriptions of rugged terrain and irregular objects, represented as elevation maps and discrete objects. It exploits diverse sensors and data sources to construct the multiple-resolution terrain maps the Ambler needs to to plan and act in the unfamiliar Martian environment.

5 Planning and Control

An autonomous planetary explorer needs to exhibit a wide range of behaviors. In particular, it must be able to navigate over a wide variety of terrain features, and to acquire many different types of samples. In addition, it must be able to detect and recover from errors in its plans, and must monitor for contingencies arising both externally (e.g., dust storms) and internally (e.g. excessive power usage). The robot should also choose plans that are "desirable," e.g., those that have a low degree of risk, high degree of reliability, and high benefit to the overall mission.

Our goal is to construct a general robot planning and control architecture that facilitates the achievement of these tasks. Consideration of the tasks indicates that the architecture should address three major issues: 1) integration of different planners, each potentially using different representations and algorithms, 2) flexibility in handling contingencies, plan failures, and unexpected situations, and 3) self-awareness of the robot's own capabilities and limitations.

The architecture we are developing addresses each of these issues. Integration of different planners will be facilitated by using a standardized declarative language to communicate between modules. Flexibility is achieved through the use of a centralized control module that will handle interrupts from environmental monitors and the scheduling of resources. The control module will also maintain dependency information between planning modules to facilitate error recovery and explanation. Issues of self-awareness will be handled using decision modules to perform such tasks as reasoning about available resources, the expected reliability and expense of a proposed plan, and the expected benefit and time criticality of achieving a given goal.

The core of the architecture is a centralized control module (see figure 10). The control module receives queries and goal requests from planning modules and routes them to be handled by applicable query or planning modules. Queries, which are used to access internal and external sensing data, are routed in a first-in, first-served manner, and the answer to the query is routed back to the requesting module. Goal requests are kept on a prioritized goal agenda, and are handled as the necessary resources become available. Prioritization, handled by decision modules, is based on the type of goal and on the current external and internal environment of the robot.

This general architecture can be used to control a wide variety of specific hierarchies of planning modules. For example, figure 11 illustrates part of our current design for a locomotion planner that integrates body and leg motions. Each box in figure 11 is a planning module, and the arrows represent the control and data flow handled by the architecture. The walk planner takes a three-dimensional ribbon and outputs a series of straight-line subgoals to be traversed. The footfall-selection-area planner (FSAP) and the footfall-location-optimization planner (FLOP) together plan where to place a foot, and the leg-recovery planner (LRP) decides how to move the leg from its current position to achieve the foot-placement goal while avoiding obstacles such as rocks and other legs. In parallel, the body-trajectory planner (BTP) decides how to move the body along the heading given by the walk planner. Its output is combined with the LRP's to coordinate movement of all joints, which is then sent to the walking-motion-manager to control the actuators.

Besides queries and goal requests, planning modules can also issue *constraints*, which are treated as advice that other modules can use to restrict their search for acceptable plans. For example, the BTP needs to constrain the FLOP, since body position helps determine the limits on the extension of the recovering leg (see figure 11). Also, a sampling planner might issue the constraint "avoid region X," which would then be used by the walk planner to eliminate region X from consideration for possible footfall placements.

When a planning module fails to generate an acceptable plan, it issues a *failure* that describes why the goal cannot be achieved. Decision modules are used to analyze the failures and to recommend appropriate action, such as reinvoking the planner with additional constraints, reinvoking a higher level planner, or using another planner that can achieve the same goal. For example, if the FLOP fails to find an acceptable area to place the foot within a given region, the architecture would reinvoke the walk planner to produce a new footfall area, adding the constraint to avoid that region.

A central feature of the architecture is the distributed nature of its problem solving. Planning, query, and decision modules can all run as separate processes and even on separate machines. Using ideas developed in the NavLab project [10], communication between modules will be transparent to the module writer, so the system can be easily reconfigured. The control module will use its knowledge of available computational and physical resources to schedule and allocate the achievement of different goals. One departure from the centralized control scheme, taken for efficiency reasons, is that query modules will have direct, high-bandwidth connections to the local terrain map and other perceptual information.

Although our aim is autonomous behavior, we realize that the Ambler will never have enough knowledge to cope with all possible situations. It is important for humans to be able to intervene and teleoperate the Ambler when the situation warrants. To this end, we are designing the control architecture to accept human input at any level of the planning hierarchy. We want to enable humans to override the goals and constraints produced by planning modules, re-prioritize the goal agenda, and override the responses of query and decision modules. These facilities will also be very beneficial during our research, since we will easily be able to substitute human input for as-yet-unwritten modules.

A major advantage of our proposed architecture is that it enables us to experiment easily with different control schemes. The openness of the architecture does not enforce a rigid discipline for constructing a robot, but does offer many tools for facilitating the construction and integration of a planetary explorer. The following three sections expand on how this architecture addresses the issues raised above of integration, flexibility, and self-awareness.

5.1 Integration

To achieve even basic competence as a planetary explorer, the Ambler will have to perform a wide variety of tasks. For example, to achieve the goal "walk over to Rock31 and check for ice on its undersurface," the robot needs the ability to plan a route to Rock31, plan and execute individual steps, including leg and body movements, position itself over the rock, grasp the rock, sense the presence of ice, etc.

For practical reasons of efficiency and ease of implementation, the planners for these different tasks need different algorithms and representations. For example, mission planning tasks typically need to be done before many relevant environmental features become known. Such tasks can be handled using largely qualitative, symbolic planning techniques (e.g., [7]). On the other hand, planning individual footfalls is a very constrained problem, and is more naturally solved using geometric and numeric algorithms.

Since each of the planning modules incorporate different representations and different assumptions about the world, a major problem is enabling the different planners to communicate effectively with one another. We plan to handle this integration problem by designing a standardized, declarative language for describing goals and perceptual queries. For example, the goal input to the leg-recovery planner consists of the desired (X,Y,Z) point to place the foot, the time interval during which it should be achieved, and constraints on the maximum and minimum allowable change to body height. The communication language is being developed by analyzing the control and data flow for the preliminary specifications of the Ambler and for the existing NavLab autonomous vehicle [10].

The use of a standard language enables implementors to agree on the interfaces between modules before actually implementing the modules themselves. Not only does this enable implementors to treat other modules as "black boxes," but it also enables modules to be substituted for one another in a transparent manner. When the control architecture receives a request to achieve a task (subgoal or query), it will have tables describing which modules are applicable for the given task. Modules for the same task can be easily substituted by changing the mapping tables. In addition, if more than one module can handle a given task, decision modules will be used to choose which one to use in the current context based on their range of expertise, computational and physical resource requirements, and expected reliability.

Another important integration issue is the handling of multiple tasks. The control architecture will maintain a list of active goals and a prioritized agenda of pending goals, and will monitor through perception when to start and stop attending to its goals. Prioritization of goals will be done dynamically, based on a cost/benefit analysis taking into account the current environment and past actions. For example, while in general it may be more beneficial to obtain sedimentary rather than volcanic samples, if a volcanic sample is nearer the Ambler might prefer to get it first, especially if it already has several sedimentary samples.

Since our architecture enables several tasks to be executed concurrently, it needs to deal with conflicts over resources. For example, the plans for moving the body and recovering leg both constrain vertical movement of the body. Our architecture will support two methods for satisfying potentially conflicting constraints. One method is to have the planners describe the allowable constraints and have a separate module resolve the constraints. This method is used by the joint-movement planner in figure 11). Alternatively, one module can produce constraints that are used by another module to restrict its actions, the method used by the body-trajectory planner to limit the FLOP's choice footfall location based on the planned trajectory of the body.

5.2 Flexibility

If the world were certain, static and benign, a robot could function with the simple control scheme of pre-planning and blindly executing the resulting plan. Unfortunately, Mars, and the world in general, is uncertain, dynamic and potentially hostile. To provide autonomous behavior, a robot must react flexibly to changes in its environment. Such changes may indicate that a plan is failing (anticipated uncertainty), in which case it must be revised, or they may indicate that some unexpected situation has arisen (unanticipated uncertainty), in which case it may be more important to achieve a different set of goals.

While the robot must attend to its sensors to detect the necessary environmental changes, practically, the robot cannot perceive all the information present at all times. Instead, it must focus on those aspects deemed to be most important. To this end, the control architecture will provide the facility to construct monitors that check on specific conditions. A monitor will specify the condition it is monitoring, the time interval over which the monitoring is to take place, the frequency to monitor, its priority for resolving resource conflicts, and what to do if the condition is found to be true.

A crucial problem is when and what aspects need to be monitored. We view this as a problem of reducing uncertainty in a plan. As time passes and actions occur, uncertainty in the robot's predicted state of the world grows, until at some point a threshold is crossed where the uncertainty exceeds the risk the robot is willing to take that its actions will succeed. This is the point where a perceptual request should be inserted into the plan, where the type of uncertainty indicates what to focus on. For example, suppose the robot plans several footfalls in advance based on its current terrain map. As footfalls are executed, the positional uncertainty of the robot grows. The robot can minimize sensing operations by checking the footfall area only when the region of positional uncertainty grows large enough to overlap with an object in the terrain map.

Noticing unexpected situations and detecting plan errors is only part of the problem. As important is handling the situation in an intelligent manner. For unexpected situations, this means deciding whether to pursue other goals and, if so, how to smoothly suspend activities for the currently active goals. For plan errors, this means repairing the plan without necessarily replanning from scratch.

The robot will handle both these contingent situations by reasoning about dependencies recorded when plans are constructed. Simply, the architecture will maintain dependency links between the inputs and outputs of a module, indicating that the decision to produce the outputs was influenced by the content of the inputs. To suspend active goals, the architecture would trace forward through the dependencies to find all subgoals, monitors, and perceptual queries that must be suspended as well.

When plan errors arise, the architecture will trace back through the dependencies to find the modules that depended on information that is no longer valid. The robot would then suspend the part of the plan that depends on the faulty information and replan from that point on [9]. This increases the performance of the robot since it does not have to replan from scratch every time an error is detected — it can reuse unaffected portions of plans, namely, those portions that do not depend on the faulty information.

Further uses of dependency information include creating explanations for human observers that describe why a particular decision was made, and in doing explanation-based learning [4] to aid the robot in avoiding similar errors in the future. Although these areas are outside the current scope of the project, the availability of suitable dependency information increases the ease with which they can be incorporated into future versions of our architecture.

5.3 Self-Awareness

Robots on Earth can be closely monitored to prevent them from performing useless or dangerous actions. For a planetary exploration robot, however, the very long signal propagation time forces responsibility upon the robot for choosing acceptable actions. To make intelligent decisions about what actions to take and what goals to pursue, the robot needs knowledge of its own resources, capabilities, and limitations in order to reason about the costs and benefits relative to other opportunities.

Costs can be measured in terms of resources used (e.g., power and time), reliability of the plan (risk), and uncertainty. Benefits of achieving a goal can be determined both by the goal's *a priori* desirability and by its marginal utility, since the benefits of achieving some goals (e.g., retrieving volcanic rocks) decrease the more they are achieved.

Decision modules will be implemented to perform cost/benefit analyses both for achieving a goal and for using a given planning modules to construct a plan for the goal. Given the goal of retrieving an object, for example, a decision module might estimate the cost of achieving it as function of the object's expected weight, its distance from the Ambler, and the expected terrain type separating the Ambler and the object. The goal's benefits might be computed as a function of the object's type, the number of like objects already retrieved, and possibly the likelihood of finding other valuable objects in its vicinity. Although initially we will use numeric measures of costs and benefits, we are also considering the use of qualitative preference measures in situations where accurate numeric measures cannot be obtained.

An additional cost that must be considered is the planning time itself. For time-critical operations, the robot must stop planning and execute the actions before it is too late. Thus, the robot needs to be aware of the deadlines for its goals, the expected time to execute its plans, and the expected time and reliability of its planning modules [2]. We are currently pursuing designs that enable the robot to reason in an efficient way about its own computational resources.

Reasoning about uncertainty in the robot's actions and its model of the world is also quite important. In the previous section we argued that explicit models of the uncertainty in the robot's actions could be used to reduce the amount of perception needed. Models of uncertainty can also be used in choosing amongst different plans to achieve the same goal. A plan with a high degree of uncertainty might be less desirable than one that is longer but more certain to succeed.

Reasoning about uncertainty can also be used to determine when to "phone home" for help. If the degree of uncertainty in a plan and its expected cost are both very high, the robot has good reason to reduce the uncertainty before venturing out on a high-risk mission. If the robot cannot use its perception to reduce that uncertainty sufficiently, it should request assistance from Earth in deciding whether to proceed with the plan.

In sum, explicitly representing and reasoning about the robot's capabilities and limitations should help it to act autonomously in an intelligent manner, avoiding undue risk and maximizing limited resources.

6 Sampling

The primary scientific purpose of the Mars mission is to acquire samples of the Martian terrain, to analyze them, and to return selected samples to Earth.[4] NASA hopes to acquire some 200 samples over a two year mission, an order of magnitude more than the Viking lander acquired during the two years it was functional. Scientists are interested in acquiring samples from a wide variety of materials, ranging from unweathered igneous rocks, sediments, and drift material, to soil, salts, and ices. The samples must be chosen with great care, however, since only about 5–10 kg is expected to be returned to Earth. A typical sample set might consist of several soil cores 1–2 m long, multiple rock cores 1–2 cm long, 5–10 soil scoops, about 100 fragments of unweathered pebble-sized rock (0.2–2 cm), and 20 fragments of weathered rock larger than 2 cm.

The sampling task for the Ambler differs in several ways from those most commonly addressed in robotics manipulation research. First, the samples will consist of granular material and irregularly shaped solids. Thus, few of the models, tactics, or effectors used by current robots to manipulate discrete, regular objects will be applicable. Second, most acquisition of samples on Mars will involve forceful interaction with the Martian surface (e.g., cutting, coring, and scooping) in an unstable, unpredictable environment. In contrast, most robotics research is involved with contact (grasping) motions in a stable (if not predictable) world. Third, the sampling operations run the gamut from coarse-grained (e.g., digging a pit to expose sub-surface features) to fine-grained (e.g., chipping off a small piece of ice). Most current algorithms are geared for tasks at a single level of detail.

These three aspects of the task greatly impact the mechanical design of the sampling equipment. Our current thinking is to design a number of simple, task-specific tools rather than one or two complex, general-purpose mechanisms. For example, rather than a six degree of freedom robot arm and hand, we could equip the Ambler with a 2 m soil drill, a 2 cm rock drill, a scooper, and a gripper.

Using multiple tools has a number of distinct advantages. First, a task-specific mechanism is likely to be simpler than a general manipulator. This implies fewer parts to break down and a simpler control algorithm. Second, by analyzing types of interactions a tool will likely encounter in performing its particular task, we can design the mechanism to survive unpredictable but expected contingencies, such as hitting a rock while coring a soil sample.

---

[4] We are concentrating our efforts only on the problem of collecting samples. The tasks of sample analysis, preservation, and packaging are outside the scope of our research.

A third advantage of multiple tools is that the range of tasks from coarse- to fine-grained can be accommodated using tools scaled to meet different requirements. For example, a soil-scooper needs to be larger and less powerful, while a rock-corer needs to be smaller and more powerful. Tools can be sized and mounted to take advantage of their power, stability, and perceptual requirements. This enables a tool's work envelope, rigidity, and payload to be maximized while minimizing weight and power requirements. A fourth advantage is reliability. If a general-purpose six degree of freedom manipulator fails, then all sampling tasks reliant on it cannot take place; if a task-specific tool fails, then only one sampling task cannot be performed.

A disadvantage of using multiple tools is that the combination might not be flexible enough to handle all possible tasks. An important area for research is to identify the range of devices needed to perform the various tasks expected to be encountered during the mission. We are also exploring alternatives for mounting the tools. One possibility is to mount them on the body and rotate the body to position and orient the tools, much like the turret tool-heads used in current CNC machines. Another, not mutually exclusive, alternative is to mount some tools on the Ambler's legs to increase stability and visibility of the sample surface (see figure 6).

Although sampling research will focus primarily on tooling configuration issues, the unique nature of the task also impacts perception and planning. Perceiving and modeling in three dimensions is complicated by the irregular nature of small-scale Martian terrain features (pebbles and surface textures). A related difficulty is predicting and modeling the destructive effects of tools on the environment — chipping off fragments from rocks changes their shape, digging pits adds new features to the terrain.

Another difficulty is that the manipulation actions can obscure perception of the sample. For example, scooping soil to dig a pit might stir up enough dust to hide the surface from a camera. We need to consider special types and positions of sensors to overcome such problems. A special case is soil coring, where traditional sensors are useless since they cannot penetrate the soil. In this case, we might use non-imaging techniques, such as measuring soil resistivity from drill torque, to gain some understanding of the sub-surface features.

On the planning side, the sampling tasks needed by the rover are all characterized by being highly unpredictable. One cannot predict when a soil corer will strike an underground rock, how a rock will fragment when struck, or where the pieces will land. Thus, even more so than for navigation, planning for sampling tasks must be highly reactive. In particular, control for functions in which sensor information is limited (e.g., clearing, digging, and rummaging) must be force- or impedance-based rather than position-based, and must react immediately to unexpected changes in force to prevent damage to the tools.

Planning techniques must also be developed to provide for a stable sampling platform. The legs and body must be positioned to maximize the leverage applied by the tools and to prevent the forces exerted from causing the Ambler to slip or topple.

Planners can exploit the coarse-to-fine range of available tools for planning sampling tasks. It is likely that hierarchical strategies will be more flexible than strategies that rely on a single manipulator technique. For example, if the mission is to retrieve an underground sample from a given region, one could imagine drilling a sample core to gain some understanding of the region before attempting to dig a pit from which to obtain a sample from the side of the pit wall.

As should be evident from the above discussions, we currently have more questions than answers about how to perform sampling tasks on Mars. Our philosophy for these problems, and for the mission in general, is to expose the difficulties and constraints inherent in the task and to use that understanding to drive the design of our mechanical, perceptual, and planning systems.

7 Current Status

Although simulations are often useful abstractions of the problems an autonomous robot will face, they are never as revealing as the actual mechanism. Our philosophy is to embed our ideas in working mechanisms that operate in natural environments. While we build the six-legged vehicle described in this article, we are testing our locomotion, perception, planning, and sampling ideas on two testbeds, both of which are currently operational.

The first testbed is a one-legged version of the Ambler. Its purpose is to begin integrating the component technologies into a single complete but simplified system that can demonstrate single-leg "walking" using a few frames of range data, simple walk planning, and simple error recovery. A full-scale leg has been built and mounted on a carriage that travels along rails on the ceiling to simulate body motion. A scanning laser rangefinder is mounted above the leg to provide data for building terrain maps. Under the testbed is a large "sandbox" containing different soil types and obstacles. A rudimentary version of the planning and control architecture, running on SUN workstations, enables the leg to lift-off, recover, and land at a chosen position. We are also using this testbed to study foot-terrain interactions (e.g., foot slippage and sinkage, power consumption during lift and footfall), and plan to add mock-ups of the other five legs to practice weaving during leg recovery.

The other testbed is a commercially available robot (a Heathkit Hero 2000) that we are using to explore ideas for combining navigation, sample collection, and teleoperation in an indoor environment. The Hero mechanism is wheeled and it has an arm and gripper. For perception, it has a base sonar and a rotating head sonar, and we have added an additional sonar on its wrist. In addition, we have mounted a camera in the ceiling of our lab to give the Hero a global overhead view. Currently, the Hero plans paths to objects using its vision system and, once in the vicinity of an object, uses its sonars to locate and grasp the object. The Hero currently picks up and deposits plastic cups and cans. Soon we expect it to retrieve printer output and to schedule the achievement of multiple, conflicting goals.

8 Summary

Constraints inherent in the task of autonomously exploring another planet have driven our design of the locomotion, perception, and planning systems for a Mars rover. In particular, the task demands a system that efficiently and reliably navigates over rough terrain, robustly perceives rugged terrain and irregularly shaped objects, and exhibits extreme self-reliance in achieving a multitude of tasks. We have incorporated all of these constraints into our design for the Ambler.

We designed the proposed six-legged walking machine to be very efficient in the type of rocky and sandy terrain expected on the Martian surface. The use of six legs enhances reliability by adding redundancy to the design. Unlike previous walkers, the Ambler should be fairly easy to control due to the decision to decouple vertical (foot) movements from horizontal (elbow and shoulder) movements. For sampling tasks, these constraints indicate the use of many simple, specialized tools, which are more robust and simpler to operate than a general arm and hand mechanism.

To achieve robust perception we employ multiple sensors, each geared to particular sets of tasks: visual data from an orbiter may be used (if available) to plan global routes; laser rangefinder data will be used to plan steps and local paths; cameras mounted on the Ambler's body will be used for selecting and guiding the acquisition of samples. To achieve efficient perception that is appropriate for different tasks we employ multiple-resolution representations: elevation maps describing the shape and properties of the terrain surfaces; complex object descriptions for discrete objects.

For planning and control, the need for self-reliance and the uncertain nature of the Martian terrain argue for a centralized control architecture to integrate new capabilities easily and to handle contingencies flexibly. The proposed architecture enables low-level procedural planners to be combined with high-level symbolic planners, using a common symbolic language of goals, queries, and constraints. Handling contingencies and plan failures is facilitated by maintaining a dynamically prioritized goal agenda and by recording dependencies between modules.

The most important requirement of a planetary rover is a high degree of autonomy. This is due both to the long signal time from Earth to Mars, and to limited bandwidth communication which makes it difficult for Earth-based observers to completely understand the robot's environment. The issue of autonomy is only partly addressed by having robust on-board perception and high levels of competence in the planners — the rover must also be aware of its own capabilities and limitations. The rover must be able to reason about the uncertainty in its models of the environment and the reliability of its effectors in planning a course of action that is within acceptable levels of risk. The rover must take its physical and computational resources into consideration in scheduling tasks. And, ultimately, the robot must know when it does not have enough knowledge and when it needs to get assistance from Earth.

By pursuing these issues of locomotion, perception, planning, and sampling in the context of a robot that operates on another planet, we hope to gain a greater understanding of the problems and solutions necessary to take intelligent machines to Mars and beyond.

Acknowledgements

This research was sponsored by NASA under Contract NAGW-1175. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies.

References

[1] J. Bares and W. Whittaker. Configuration of an Autonomous Robot for Mars Exploration. In *Proc. World Robotics Conference*, Society of Mechanical Engineers, Gaithersburg, Maryland, To appear, May 1989.

[2] T. Dean and M. Boddy. An analysis of time-dependent planning. In *Seventh National Conference on Artificial Intelligence*, St. Paul, Minnesota, 1988.

[3] M. Hebert, T. Kanade, and I. Kweon. *3-D Vision Techniques for Autonomous Vehicles*. Technical Report CMU-RI-TR-88-12, The Robotics Institute, Carnegie Mellon University, 1988.

[4] T. Mitchell, R. Keller, and S. Kedar-Cabelli. Explanation-based generalization: a unifying approach. *Machine Learning*, 1, 1986.

[5] M. Raibert. Introduction to special issue on legged locomotion. *International Journal of Robotics Research*, 3(2), Summer 1984.

[6] J. R. Randolph, editor. *Mars Rover 1996 Mission Concept*, Jet Propulsion Laboratory, Pasadena, California, 1986. Available as Technical Report D-3922.

[7] E. Sacerdoti. *A Structure for Plans and Behavior*. American Elsevier, New York, 1977.

[8] S. A. Shafer. *Automation and Calibration for Robot Vision Systems*. Technical Report CMU-CS-88-147, Carnegie Mellon University, May 1988.

[9] R. G. Simmons. A theory of debugging plans and interpretations. In *Seventh National Conference on Artificial Intelligence*, St. Paul, Minnesota, 1988.

[10] C. Thorpe, M. Hebert, T. Kanade, and S. Shafer. Vision and navigation for the carnegie-mellon navlab. *Annual Review of Computer Science*, 2:521–556, 1987.

[11] B. Wilcox and D. Gennery. A mars rover for the 1990's. *Journal of the British Interplanetary Society*, 40:484–488, 1987.

[12] D. D. Wright and R. E. Watson. Comparison of mobility system concepts for a mars rover. In *SPIE Vol. 852 Mobile Robots II*, pages 180–187, Society of Photo-Optical Instrumentation Engineers, Cambridge, Massachusetts, November 1987.

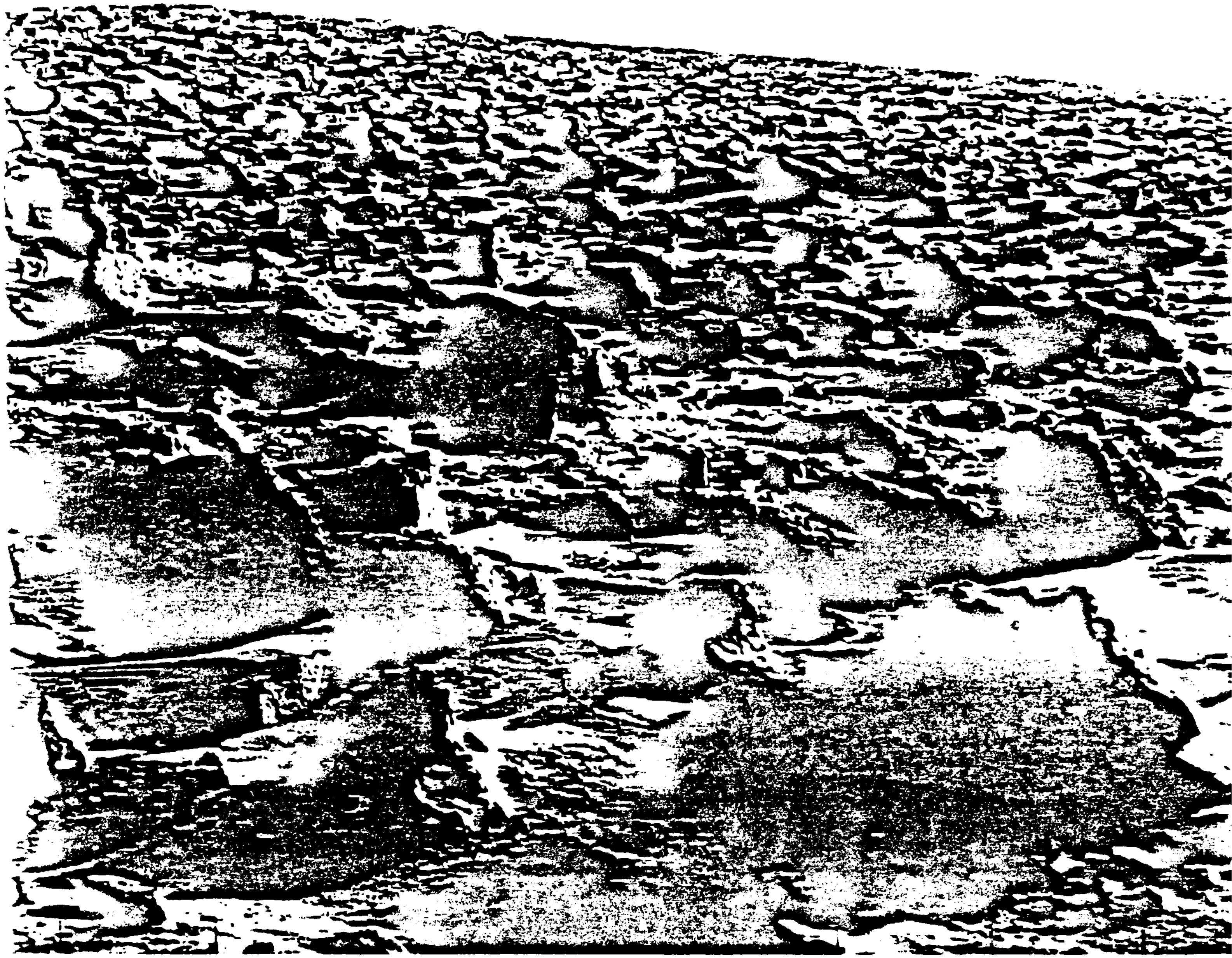


Figure 1: Martian Terrain Viewed by the Viking 2 Lander

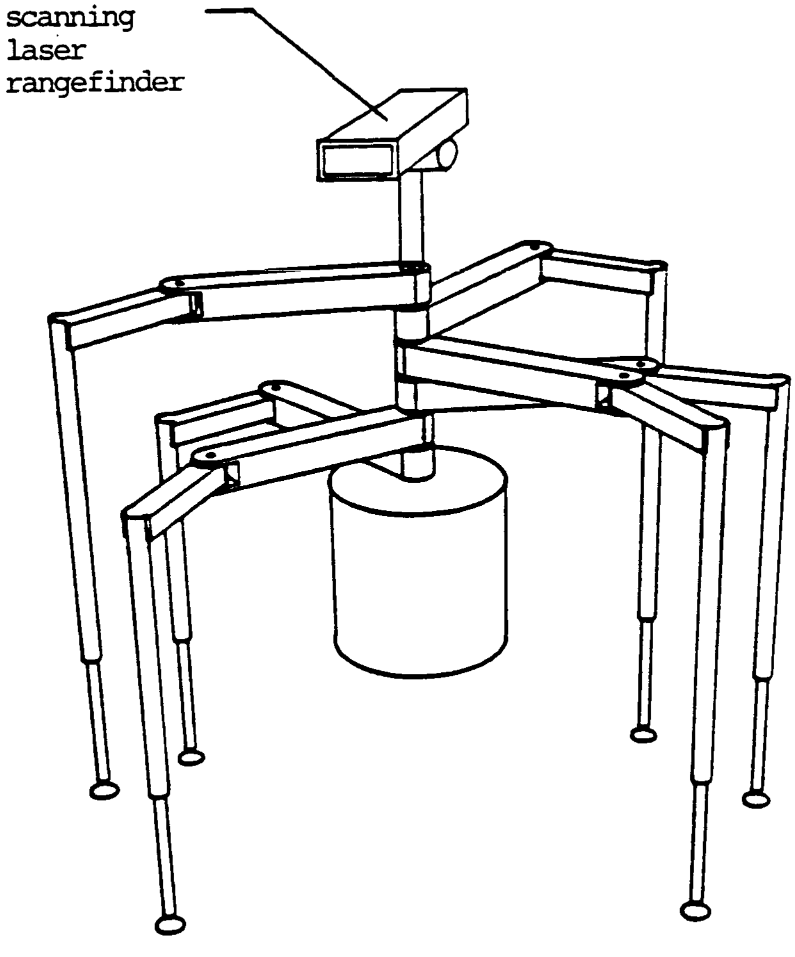

scanning
laser
rangefinder

Figure 2: Sketch of Ambler

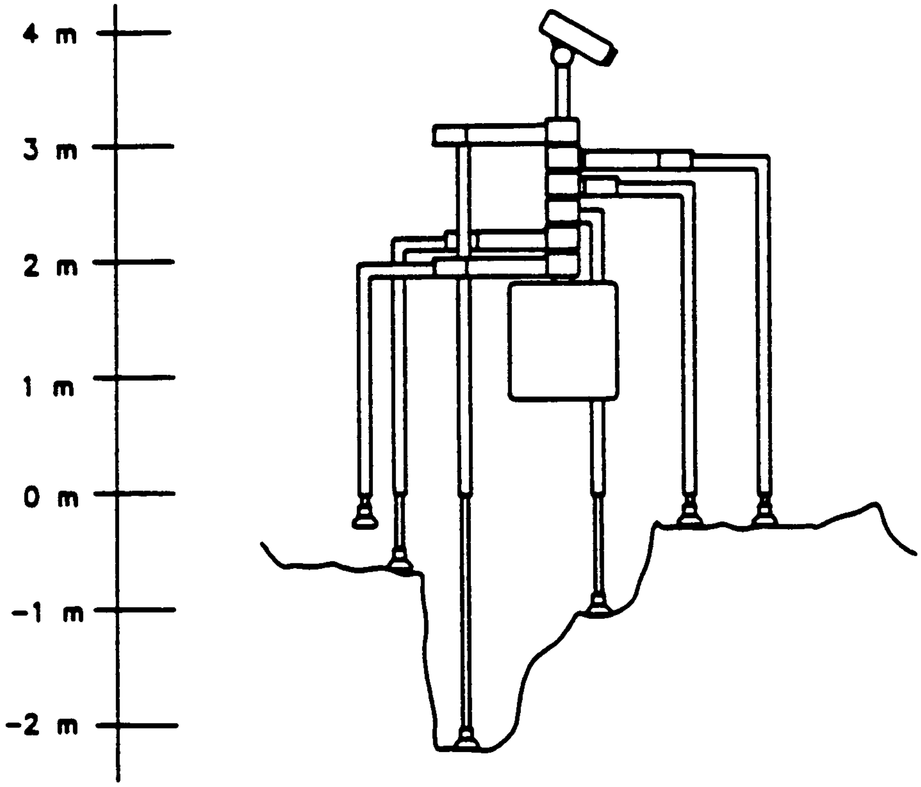

4 m
3 m
2 m
1 m
0 m
-1 m
-2 m

Figure 3: Leg Stack

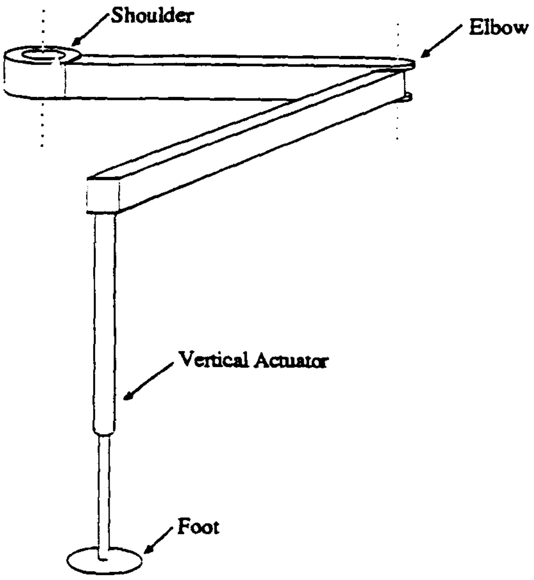

Shoulder
Elbow
Vertical Actuator
Foot

Figure 4: Side View of Leg

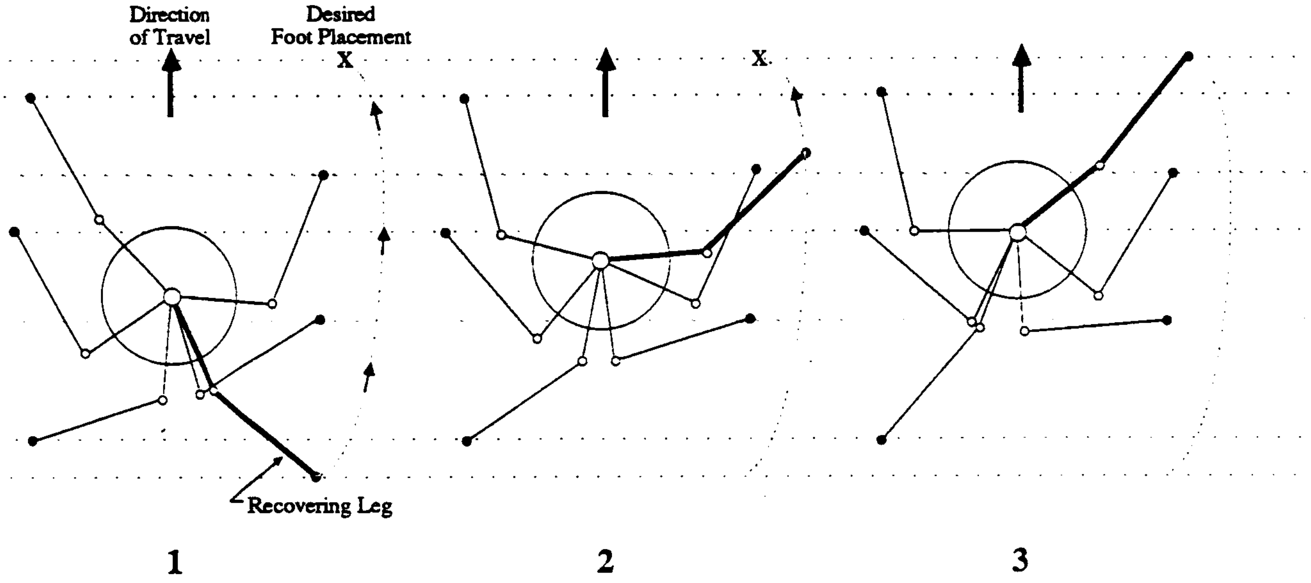

Direction of Travel
Desired Foot Placement
X
X
Recovering Leg
1
2
3

Figure 5: Overlapping Gait
As the walker advances, the recovering leg (drawn darker) overlaps the two right side supporting legs (dashed line). Depending on a leg's location on the central stack, some weaving around supporting legs may be necessary for it to recover past forward legs.

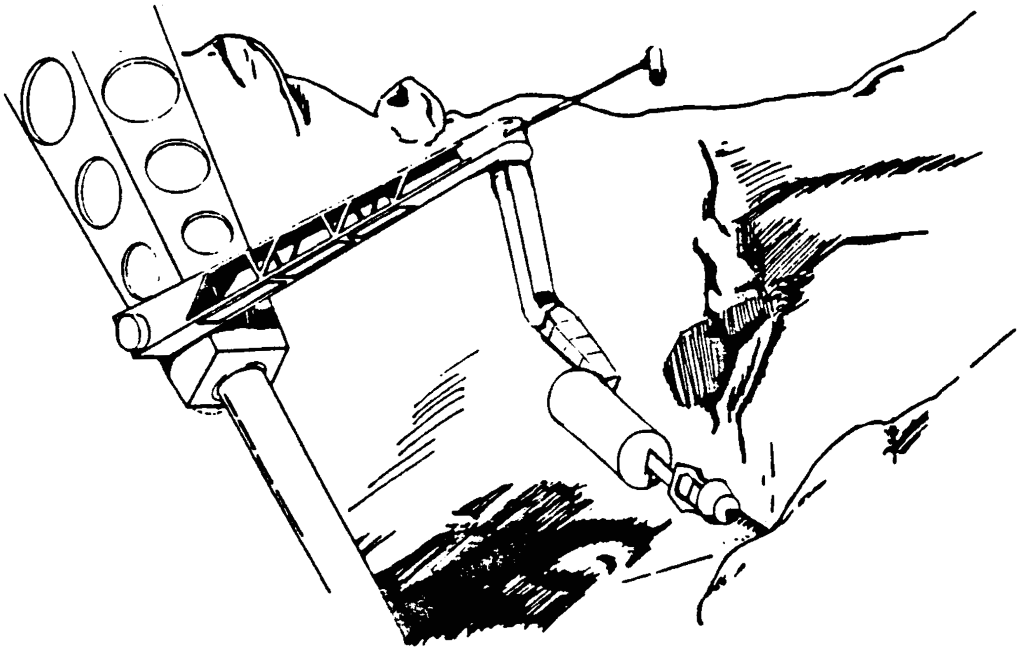


Figure 6: Leg-Mounted Sampling Tool

This is a somewhat fanciful artist's conception. The leg should be vertical, and the "arm" holding the drill should be less articulated.

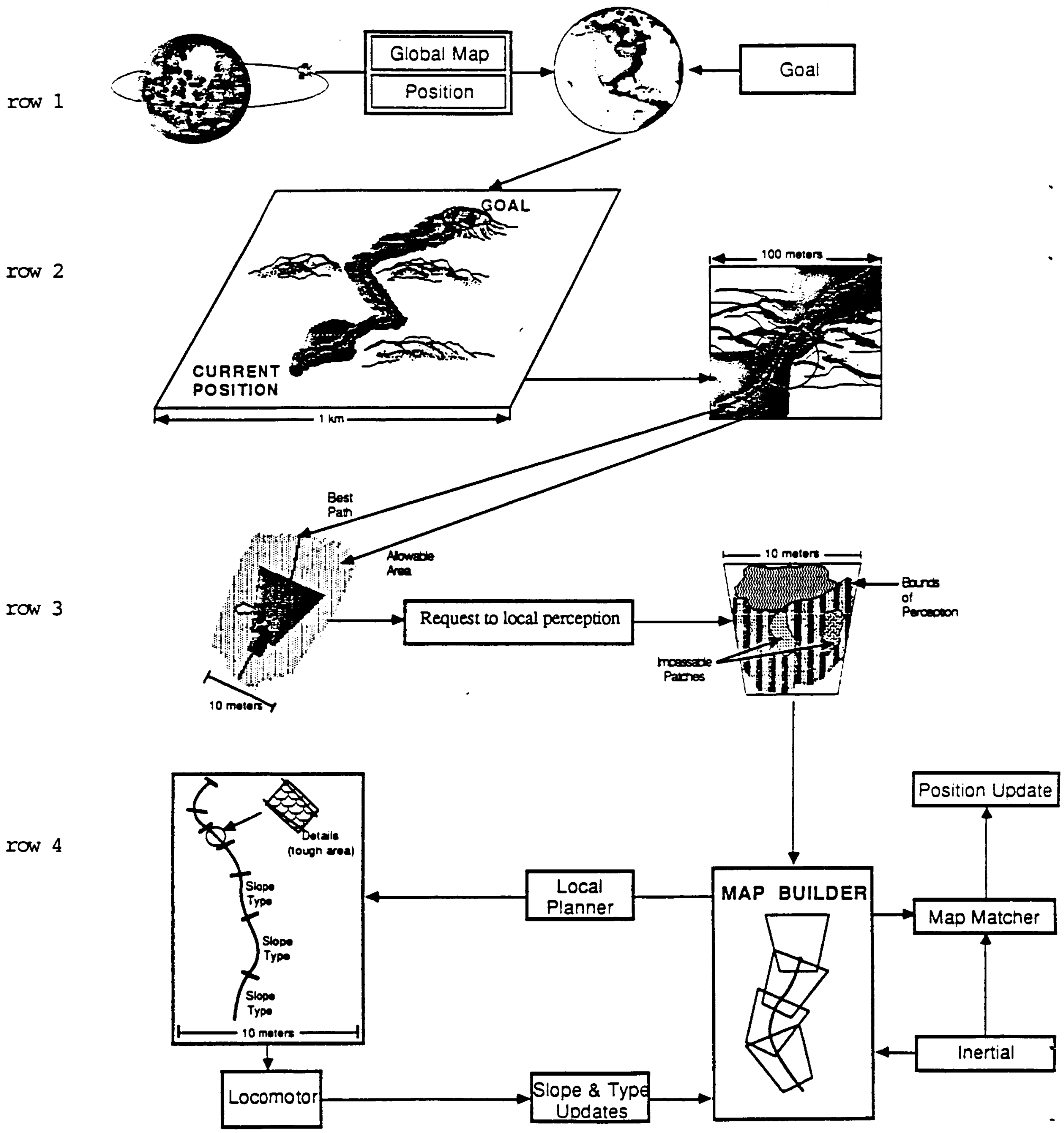

row 1
Global Map
Position
Goal
row 2
GOAL
CURRENT POSITION
1 km
100 meters
row 3
Best Path
Allowable Area
10 meters
Request to local perception
10 meters
Bounds of Perception
Impassable Patches
row 4
Details (tough area)
Slope Type
Slope Type
Slope Type
10 meters
Local Planner
MAP BUILDER
Position Update
Map Matcher
Inertial
Locomotor
Slope & Type Updates

Figure 7: Map Resolution Levels

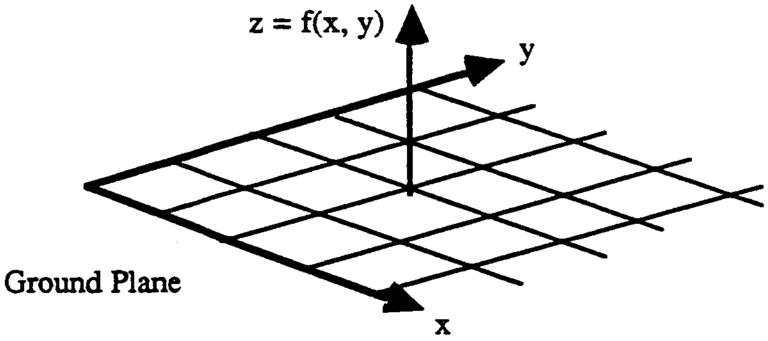

z = f(x, y)
y
Ground Plane
x

Figure 8: Elevation Map Structure

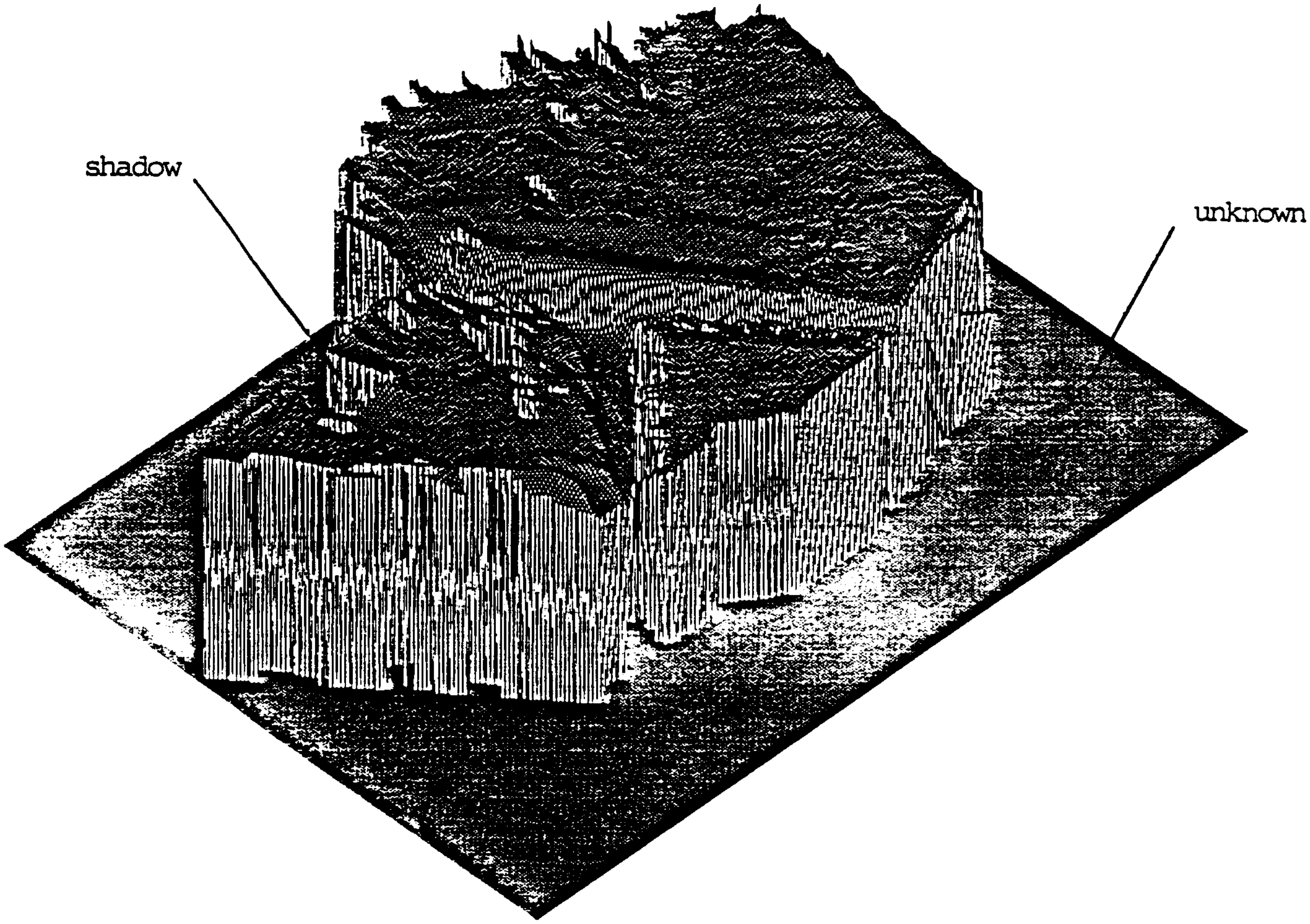


Figure 9: Example of Elevation Map

This figure is an example of a composite elevation map constructed by merging four rangefinder views of the rugged terrain at a construction site. The grid size is 10 cm.

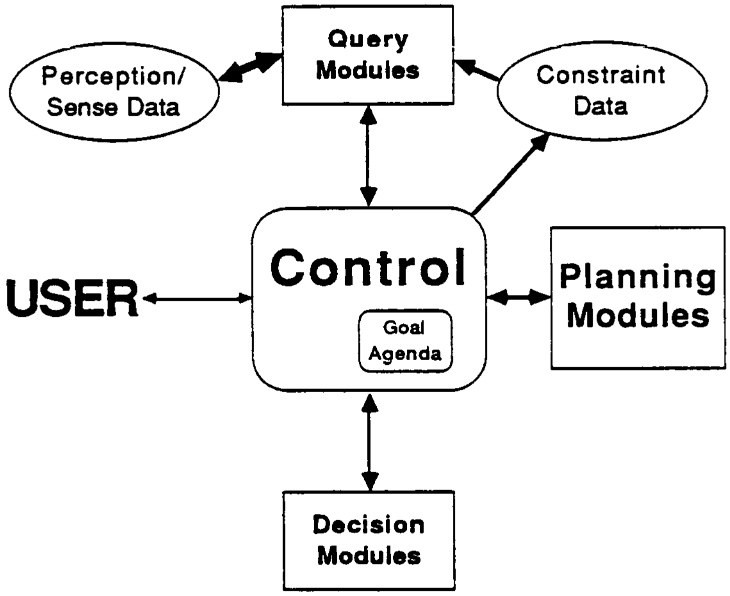

Query Modules
Perception/ Sense Data
Constraint Data
USER
Control
Goal Agenda
Planning Modules
Decision Modules

Figure 10: Planning and Control Architecture

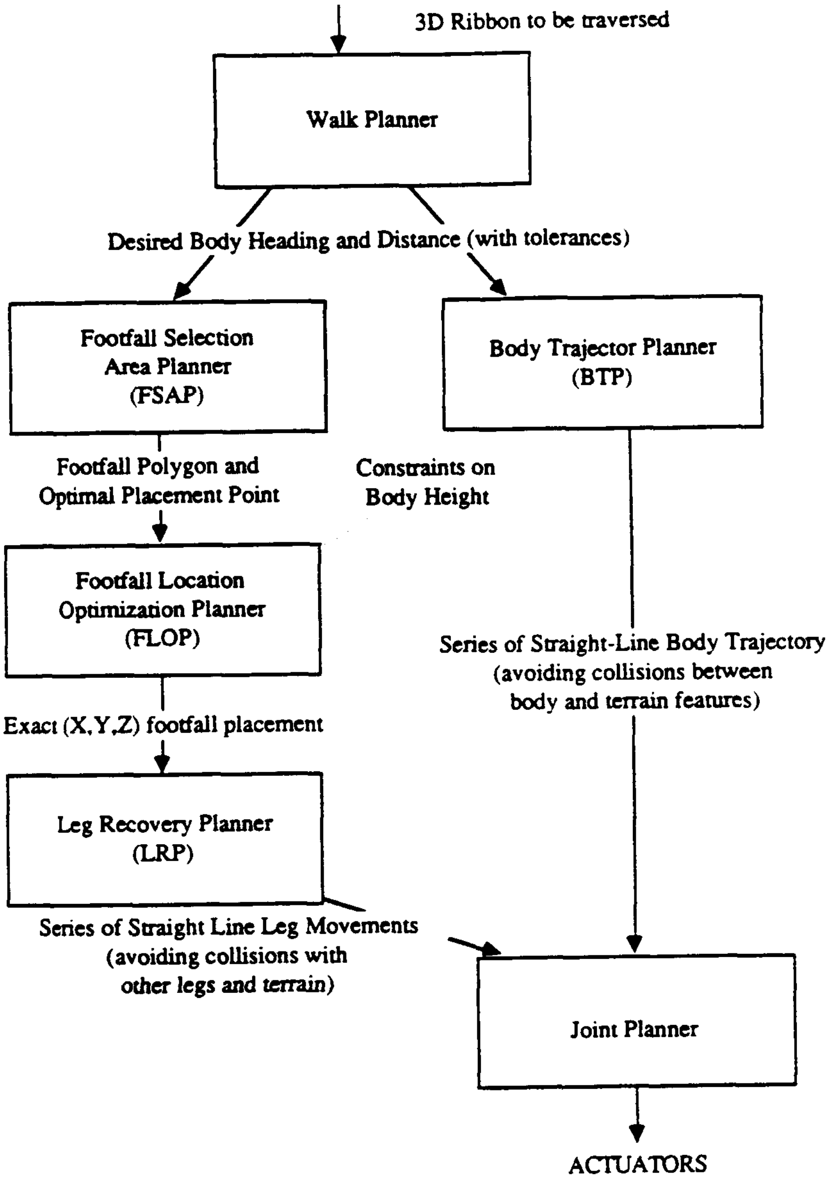

3D Ribbon to be traversed
Walk Planner
Desired Body Heading and Distance (with tolerances)
Footfall Selection Area Planner (FSAP)
Body Trajector Planner (BTP)
Footfall Polygon and Optimal Placement Point
Constraints on Body Height
Footfall Location Optimization Planner (FLOP)
Series of Straight-Line Body Trajectory (avoiding collisions between body and terrain features)
Exact (X,Y,Z) footfall placement
Leg Recovery Planner (LRP)
Series of Straight Line Leg Movements (avoiding collisions with other legs and terrain)
Joint Planner
ACTUATORS

Figure 11: Partial Planning Hierarchy for Locomotion

Configuration of an Autonomous Robot for Mars Exploration

John Bares
William Whittaker

Field Robotics Center
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

(412) 268-6556

1. Introduction

Space programs are targeting manned exploration of Mars; however, manned missions elude financial and technical feasibility. Mobile, perceptive robots hold the prospect to explore Mars on behalf of man. The challenge motivates a robot with unprecedented ability for the autonomous, self-reliant exploration of rugged, barren terrains. This paper presents the configuration for such a robot.

A robot configuration is the comprehensive functional design from which detailed design and production of a prototype robot system follow. Configuration is the critical milestone of successful robot development. The configuration and eventually the design of an autonomous robot are driven by the functional specification of the task. Functional specifications are generally implied by the task statement or derived from subjective consideration of the end solution requirements. Functional specifications evolve to satisfy the needs of configuration and design.

The scope of configuration of an autonomous exploring robot is comprehensive, incorporating mechanisms, computing and control hardware, and processes for perception, planning and action. The process of perception acquires, interprets and models sensor data about the environment. Planning operates on this model to prescribe robot motions toward accomplishing goals. With a consideration for robot self-preservation, the action process implements planned motions.

In this paper we discuss configuration of a robot, the *AMBLER*[1], for autonomous exploration of the rugged terrain of Mars. We first give an overview of the AMBLER, with attention to mechanisms and processes for perception, planning and action. The AMBLER configuration is then justified from the task functional specifications and competing needs of interacting subsystems.

---

[1] AMBLER is an acronym for Autonomous, MoBiLe, Exploration Robot, meaning one that moves about from place to place at an easy walk with a sauntering gait.

2. Mars Exploration

Mars has more Earth-like features than any other body in the solar system. Mars is geologically interesting, featuring volcanos, canyons and dry river beds. It provides the opportunity to learn how geology is affected by pressure, gravity and temperature conditions different than on Earth. An exploration robot can collect various samples and perform on-site analysis of environmentally sensitive materials (e.g., ice). It can also document geology, including lava flows, rock, glaciers and river valleys.

Mars Rover/Sample Return (MRSR)[1] is a NASA mission to fly a mobile robot to Mars to conduct scientific and sampling activities. After a 2-year traversal of several hundred kilometers the robot will find and dock with another spacecraft that will shuttle the samples back to Earth. Severe limitations on the sample payload that can be returned to Earth require a sophisticated, Mars-based exploration, sample evaluation, selection and preservation program. The total quantity of samples expected to be returned is only 10's of kgs. All the intellectual and technical resources of the Earth's scientific community can be brought to bear on the returned samples. The samples (e.g., pebbles, rock chips, cores, drillings) will provide direct information on the composition of the planet's crust, mantle and core, and on geologic and climatic history. The broad objectives of MRSR are thus to gather as much scientific information as possible and to return to Earth an intelligently-selected sampling of materials.

The half-hour signal delay in round-trip telemetry from Earth to Mars precludes teleoperated (human-removed) control due to the extremely slow progress that would result and forces robot autonomy. Autonomous operation implies that the robot is able to operate in isolation without failure, damage or entrapment with only occasional input from a human overseer. Exploration differs significantly from operations in conditions that are known or predictable. By nature, exploration defies preplanning, at least at the grain of the local environment, so a robot's success relies heavily upon its abilities to perceive its state and surrounding environment, plan and execute actions and survive. Self-reliance is the single most important consideration that drives the configuration of an autonomous Mars exploration robot. Any debilitating circumstance results in mission failure.

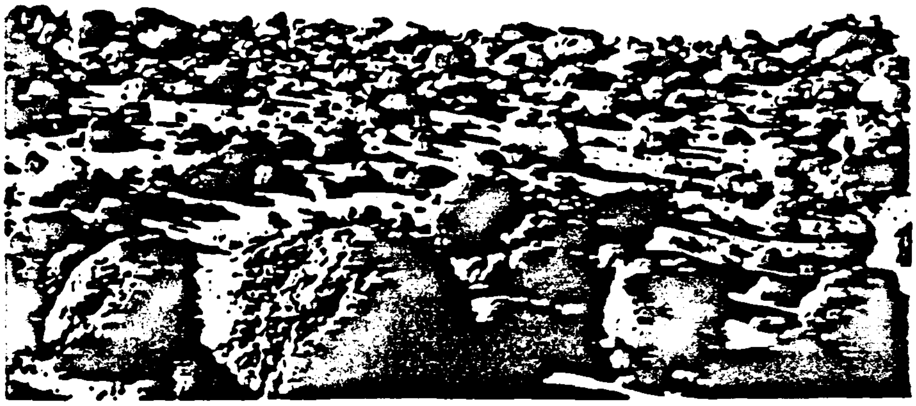


Fig-1: Surface of Mars

With mass restricted to hundreds of kilograms and power restricted to hundreds of watts, a Mars exploration robot is challenged to successfully navigate on terrain – typified by the dense boulder field with interspersed dunes shown in Fig-1 – at an approximate rate of 1 km/day for several years. Boulders, crevasses and ledges of one-meter scale are expected in the three-dimensional, discontinuous Martian terrain, as are very low bearing strength soils. Traversal of such terrains requires capabilities of modeling, planning and locomotion that go beyond those which succeed in two-dimensional environments such as a smooth factory floor. Furthermore, unlike typical industrial settings, Mars exploration has many competing objectives including power efficiency, stability and rate of progress. The need for extreme self-reliance is complicated by rugged terrains; for many locomotion mechanisms, traversal of difficult terrain compromises safety and self-reliance. Environmental factors such as low temperatures, winds and dust impose additional constraints.

Barren terrains, whether terrestrial or planetary, are all rugged natural surfaces devoid of structured content like buildings or trees; they are vast landscapes of rock, soil and ice. The surface of Mars is dominated by diffuse, granular media and irregularly shaped, solid natural forms. In comparison, the surface of the Earth's Moon is quite smooth and firm. The absence of vegetation, man-made structures and animate objects on barren terrains significantly influences autonomous robots because, on one hand, they need not cope with the variety of associated impediments to sensing and motion; on the other hand, a robot cannot rely on easily identified features like trees or fixed structures as landmarks for navigation aids. Of interest are terrain attributes such as geometry and material mechanics that directly influence locomotion, sampling and self-preservation.

A robot configuration is the functional design that derives from the specification of the robot's environment and purpose. For the barren, rugged terrains of Mars, a configuration is forced to address three-dimensional, irregular morphology and mechanical terrain behavior. The purpose of exploration requires that the configuration incorporate mechanisms to traverse rugged terrain, make observations and gather samples. The inability to preplan tasks and trajectories requires that the motions of an exploratory mechanism decouple to simple, assessable (easy to measure) actions. These motion primitives must be power efficient and power expenditure must be modeled, costed and controlled by the configuration. Autonomy and self-reliance require that primitive motion and interaction models be robust and tractable. The robot must have abilities for self-diagnosis, contingency action and safeguards beyond any precedent.

3. AMBLER: An Autonomous Robot for Mars Exploration

This section overviews the AMBLER, a robot configuration to autonomously explore the rugged surface of Mars. The AMBLER [Fig-2], a six-legged walking robot, perceives and models terrain, and plans and executes tasks and motions; it is unmanned, self-contained and power efficient. A predominant philosophy underlying the AMBLER configuration and its operation is extreme self-reliance, manifested primarily by highly predictable mechanisms and conservatism at all levels of planning.

The AMBLER uses a laser range scanner to gather local terrain data. This data from varied view perspectives, sensor types, resolutions and times is processed to generate an elevation map, an effective representation for rugged terrain. The elevation map is a grid plane that is conceptually fixed to the terrain and is indexed or "scrolled" ahead occasionally as the vehicle moves through the terrain. Each grid point contains data such as terrain elevation and compliance. Grid points may also contain derived attributes such as terrain slope and curvature. Multiple images are merged onto the elevation map, accounting for error in robot motion, sensor noise and incomplete sensor coverage. Data confidence increases as new data supports prior readings.

Continuously updated robot state models include world position, position with respect to the elevation map, joint positions and sensor orientations. Some state representations such as world position have an associated uncertainty. Given the local terrain map, robot state models and other on- and offboard inputs, AMBLER planning decides what actions will further progress.

Several different planners formulate behavior of the AMBLER from navigation on the Martian surface to selecting foot placements and sampling motions. Conceptually, one set of planners determines "task" plans; the second set plans "motion sequences" that forward the desired task intentions. Throughout planning, competing objectives must be considered including energy expenditure, stability and rate of progress. In a simple example, a task planner first denotes a locomotion task of travel to a distant terrain point. Several levels of motion planners then determine a sequence of body, leg and foot motions that will attain the desired goal while adhering to a given stability margin and maximum energy level. Motion sequences (trajectories) resulting from motion planning are backsolved to motion primitives and queued for execution by the mechanism. Conditions expected to occur in the course of execution are anticipated by planning for use in monitoring the acceptability of the resulting AMBLER actions. It is necessary that planners preclude tipover at all costs, as it is unlikely that fragile sensors and antennas could survive the impact of such an event. One method of providing this safeguard is to plan so that no action will be taken without some caution for the unexpected.

The AMBLER locomotion mechanism has six legs to isolate the body from terrain and to efficiently propel the body. Each leg has two revolute motions in the horizontal plane that position the leg over the terrain, and a vertical telescoping motion that extends the foot into contact on the ground. Each AMBLER leg is mounted at a different elevation on the central axis of the body and can rotate fully around the body. The vertical links in the AMBLER's legs level the body over terrain in the manner of a raft floating on water over a rough lake bottom. Propulsion of the level body, which requires only several of the planar revolute motions, is analogous to poling the raft. As legs reach the limit of their stroke, they are replaced ahead of the walker much like the pole is replaced ahead of the raft.

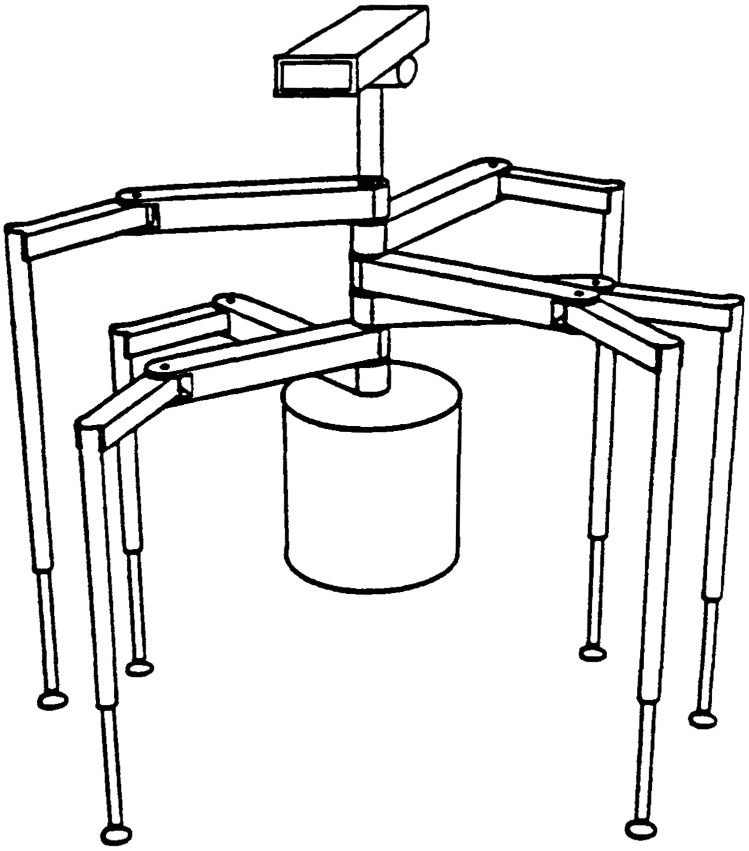


Fig-2: The AMBLER

While the AMBLER can vary its height and width to compact, its average overall height, from ground to the top of the leg stack, is ~3.5 m and normal walking width is ~3 m. The AMBLER can cross 1 m obstacles while maintaining a level body attitude. The laser range sensor used to build the terrain elevation map is mounted above the leg stack for a good view of terrain. Communication equipment and scientific sensors are also mounted above the leg stack.

The AMBLER body, a 1 m dia. cylinder situated below the leg stack, contains power generation, computation, sampling equipment and scientific instrumentation. Scientific observation and sampling needs exploit the benefits of positioning and reaction that the body motions can provide: Since the body can hover to station over desired sampling locations, only light, short sampling motions (e.g., drills, small grasping implements) need to be appended to the underside of the body. Doing so reduces the number of sampling motions and the associated needs for volume, payload, power and control. The result, including body motion, is a manipulator chain from large-to-small and from coarse-to-fine motions. High resolution terrain sensors are configured under the body in a down-looking manner in support of underbody sampling equipment. Because of their large size and long reach, legs might also be used to house and position additional sampling and scientific equipment. For instance, due to the long length of a 1 m core drill assembly, mounting the mechanism on the vertical axis of a leg appears beneficial.

The AMBLER computing system operates processes for perception, planning, motion control, communication and health monitoring. 32-bit processors configured on a common backplane are the core of the computing system. The computing system configuration includes the ability to remotely program, debug and test algorithms. Where response to asynchronous external events is critical to the survival of the robot, the configuration is streamlined to enable timely throughput.

AMBLER software is object-centered; that is, software objects (sections of code that perform a specific task) are viewed as functional units that receive and send data/commands directly from/to other objects. The computing system permits human interaction with most objects. Ultimately, human input will be in the form of occasional task objectives, though during development of the prototype system, continuous teleoperation and direct motion control (e.g., servo-level) are likely.

4. Description of the AMBLER Configuration

A configuration is a complete functional design that directly generates a detailed design and ultimately a physical robot. Configuration of a Mars explorer must thoroughly consider the elements of perception, planning, action, computation and infrastructure. The objective is to work from functional specifications to configure elements that are compatible, complementary and contribute to a composite functional capability. A configuration must take full advantage of elemental technologies and at times go beyond to forge the technologies into a needed functionality or performance. The key to a successful configuration is proliferation of inter-elemental complements.

This section presents the content of major AMBLER subsystems including local terrain perception, motion planning and locomotion mechanism. The scope of this paper precludes the treatment of other subsystems including intermediate and global perception, task planning, motion control, science/sampling, computing hardware, health monitoring, communication and infrastructure.

4.1. Local Terrain Perception

Detailed, timely information about terrain geometry and composition is generated by the AMBLER explorer. Local perception builds and maintains descriptions of terrain surrounding the robot in support of planning for locomotion, self-preservation and autonomous sampling. Geometric information about the terrain (e.g., surface location, slope, curvature) is used by the AMBLER to generate motion plans. This section discusses the terrain representation and how it is built from range data.

Terrain Sensing

Range sensors are the preferred sensing mode for building local terrain models. Range data directly measures geometry, which is the principal content of local terrain models. Passive vision ranging techniques (e.g., stereo), though power efficient, are ineffective in segregating the weathered geological materials of Mars. Soils, sands and geologies of Mars appear rather the same to a camera. This blandness disempowers traditional techniques of passive vision that rely on distinctions of color or intensity. In such barren terrains, passive vision must consider second order cues like interpretation of shadows and spectral surface reflection.

The AMBLER will employ a two-axis laser range scanner. The approximate field of view is a 60° cone emanating from the scanner. Assuming a 3.5 m mounting height (top of AMBLER leg stack) and steep downlooking angle, resolution for the first 5 m is expected to be 3-5 cm and decrease to ~30 cm at the scanning limit of 15-20 m. These resolutions and relative ranges fulfill the various modeling requirements for locomotion and science/sampling. In addition to range, the scanner provides a surface reflectance reading that can be an indicator of surface material.

Laser range scanners require power for ranging and for scanning. In the interest of power economy, a possible class of range sensor might forego the luxury of two-axis scanning, using robot progress or appendage motions to accumulate complete terrain coverage. For example, a conical shroud of range data might be acquired with a single, narrow-beam laser ranger and a single rotary scan motion. Deployed high on a mast, such a sensor would cut out a dense locus of range data on a wave front that moves with the robot, relying on accumulation techniques to build and retain a full terrain model.

Terrain Representation

The foremost content of a terrain representation is the accurate geometric model of the highly irregular surfaces common to rugged terrains. While geometry is the single most important feature, other terrain attributes including material type and compliance are required in the representation for assessing robot-environment interaction.

The terrain model must clearly identify areas that haven't been sensed because they are out of scanning range (unknown) or shadowed by a terrain feature (occluded). Knowledge that a region is hidden from or beyond sensor view is important to a robot that explores. The representation must merge sensor images from different vehicle positions and view perspectives to enhance quality and infill unknown and occluded areas. Finally, costly data transforms to change reference frame or data form should be minimized.

Surface patch representations have successfully modeled the "uneven" terrains navigated by the CMU NavLab[2]. Surface patch representations model a surface as a mosaic of many smooth connected areas or patches – the density and size of which depends on local surface roughness. The process of building a surface patch representation requires image segmentation, calculation of surface normals and curvature, and finally, fitting of planes or quadrics. An immediate drawback is that computational requirements increase significantly with terrain roughness (many more approximating surfaces are required). Furthermore, the walking locomotion planner requires terrain elevation to select foot placements – another approximation is therefore required to backsolve elevation from a patch representation.

Cell Models

Cell representations subdivide a three-dimensional region into large numbers of small cells, and sensed attributes such as range and color are accumulated into these cells with attention to noise and uncertainty. Cell models can be efficiently post-processed to derive geometric features such as slope and curvature. A 3-D cell model subdivides the local world into equally-sized cubes; cubes on or below the terrain surface are labeled "occupied", and cubes above the terrain are labeled "empty". Other terrain attributes such as color and compliance can also be tagged to a cube. Cubes are sometimes termed "buckets" because their data representations conventionally hold any number of attributes and corresponding uncertainty values. An elevation map is a 2.5-D cell model where each grid point is a bucket containing, at minimum, an elevation value.

The cell model is a uniform means of consolidating data from various types, resolutions and view perspectives of sensors over temporal and motion histories. Cells can accumulate information about terrain color, geometry and material attributes like friction angles, density and cohesion, all of which are important to robots in rugged terrain. Because of their inherent ability to model irregular surfaces, cell representations are well suited to represent the detail of amorphous shapes typical of the rugged Mars terrain.

Cell models recursively subdivide volumes to arbitrarily fine grain size. At any resolution, data can be accumulated into or accessed from the model. By averaging data values or interpolating between data points, resolution in any region of a cell model can be increased or decreased to suit the needs of planning. This innate ability of cell models to collapse from coarse resolution to fine resolution in a local region supports, in a consistent context, the diverse grains of planning from traveling a valley to grasping a pebble.

Cell models are ideal representations for casting notions of emptiness, occupancy and invisibility. In rugged terrains where boulders may occlude important portions of a scene, information about what is visible generates the dual information of what is invisible. Cells with unknown or occluded values are flagged to enable straightforward detection by planners.

If a majority of terrain points have only a single elevation value, as is the case with barren terrains, 2.5-D cell models are more computationally efficient than 3-D cell representations. A 2.5-D cell model is especially amenable to planning AMBLER locomotion as selection of foot placements reduces to planar searches under constraints such as maximum elevation and slope. Similar benefits of the 2.5-D representation are predicted for planning sampling motions. A drawback of the 2.5-D representation is that it is difficult to represent terrain points with multiple elevations (e.g., vertical and overhanging surfaces). It is, however, possible to detect and flag the occurrence of such areas in an elevation map.

2.5-D elevation maps can either be generated from 3-D cell models by tracing up cube columns and concatenating columns when the terrain surface is reached[3] or constructed directly from range data. To minimize data manipulation we chose the second approach, that is, to build a 2.5-D cell model directly from sensed range values. The result is that raw sensor data is processed minimally to build the elevation map.

Local Terrain Map (LTM)

The AMBLER elevation map, termed the Local Terrain Map [Fig-3], is approximately 30 x 30 m and is divided into sections of high and low resolution: ~5 cm resolution for the first 5 m and ~30 cm resolution for the next 25 m. The Local Terrain Map (LTM) is conceptually tied to a terrain position in the vehicle locale; the AMBLER moves with respect to the LTM. The LTM is occasionally moved ahead (scrolled) and re-correlated to the vehicle position to eliminate accumulated error due to vehicle locomotion error. Map grid points contain various types of information including elevation, uncertainty, compliance, color, derived terrain features such as slope and curvature, and flags denoting unknown and occluded areas. As each new frame of range data is taken it is merged to the LTM.

The local terrain map is "site-fixed" to a terrain position in the locale of the vehicle. To build a quality representation of the local terrain, the map must merge several range images. There is a choice to fix the map either to the vehicle (e.g., body-centered) or to a point on the terrain (site-fixed). Implementing the former means that new images are directly merged to the map without transformation, but the map must be advanced continuously as the body moves. The latter requires that each new image is transformed to the site coordinates of the map and merged and that the map is scrolled forward only occasionally.

Based on relative sizes between sensor footprint (scan area) and the LTM, each new range image contributes roughly 20% new information to the map. Therefore, keeping the body-centered map current (i.e., moving it with the body) would require approximately 5 times as many transformation computations as the alternative of transforming the new image back to the site coordinate frame. Depending on vehicle movement, a data transform in either case *could* mean 3 translations and 3 rotations for each grid element. Beyond computational efficiency, a site-fixed map also seems appropriate when the AMBLER is to map a prospective sampling area; a single site-fixed terrain map could accurately hold sensory data for an extended period of mapping activity and small vehicle motions.

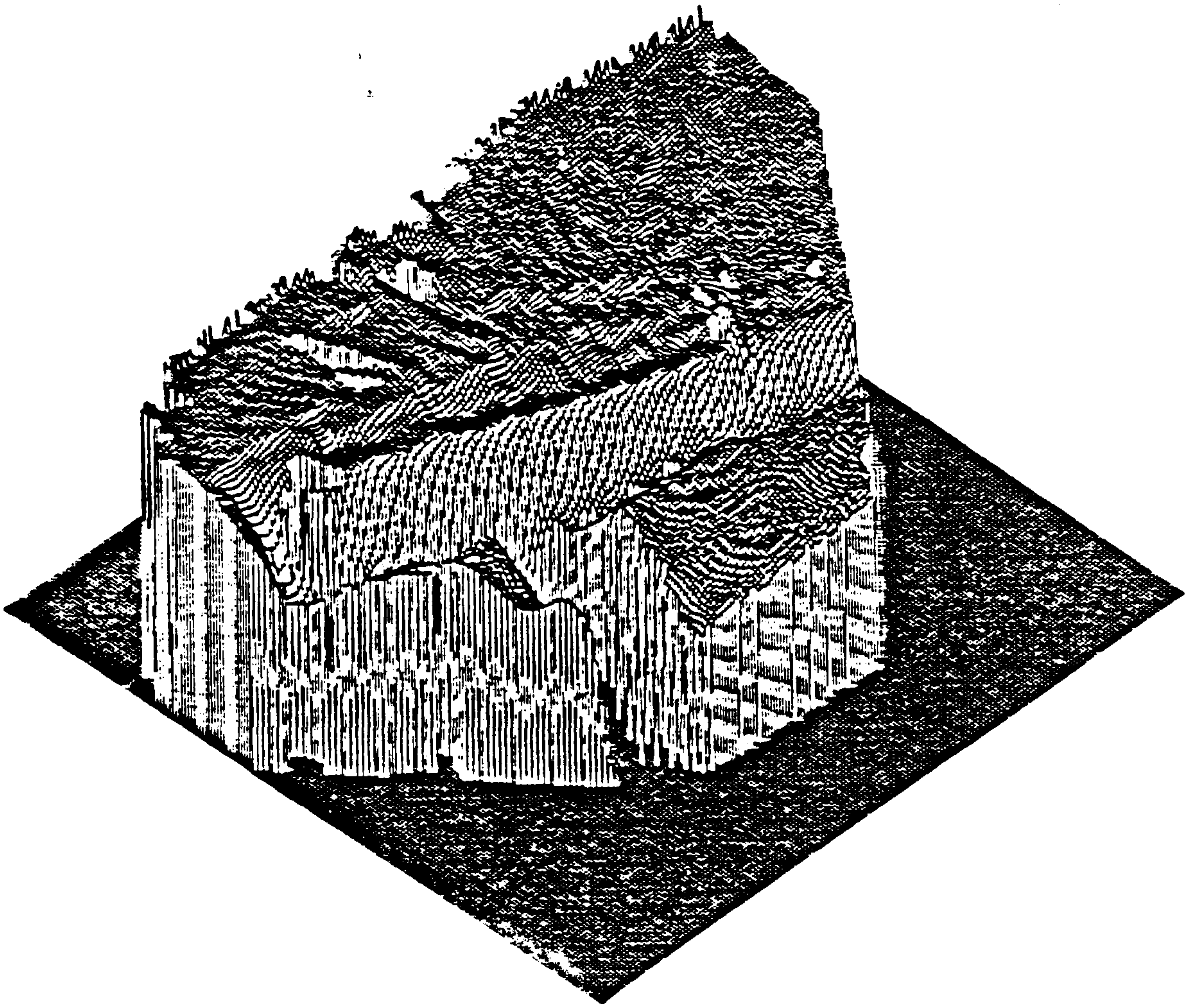


Fig-3: Local Terrain Map[4]

When merging a new range image to the map, the match can be performed "blindly" by incorporating only dead-reckoning (internal motion assessment) information from the locomotor. Depending on the reckoning accuracy of the locomotor – the AMBLER is predicted to be very accurate – blind matching could introduce error into the LTM. If so, a high-resolution matching algorithm (e.g., iconic) can be used, at the expense of significant computational burden, to determine a nearly exact match. In either case, the LTM and current vehicle position can be occasionally re-correlated by scrolling the map.

Because of its scale relative to the terrain features expected on Mars, it is possible that the AMBLER locomotor will be able to traverse a majority of the intended terrain with straight-line body motion. Additional benefits accrue for special instances when the robot moves in rectilinear motion consistent with the axes of the site-fixed LTM; the costly 6-degree-of-freedom (DOF) transformation required to merge a new image is reduced to a fast 1-DOF transform.

4.2. Motion Planning

Robotic exploration of the barren unknown terrain of Mars defies detailed preplanning, so an exploration robot must generate its own motion commands, execute them and respond to the resulting consequences. Planning in rugged terrain represents a significant departure from robot planning to date by requiring rigorous consideration of the three-dimensionality of terrain, mechanism interactions with the terrain and multiple competing objectives such as minimal energy expenditure and maximum mechanism stability. In addition, planning must provide safeguards and respond to contingencies – such as weak soil conditions – that inevitably befall an unmanned agent.

This section discusses configuration of the AMBLER motion planning subsystem. Planners that supply task objectives to motion planners are not discussed; these are viewed to operate at slow rates and their function may be fulfilled in part by human input from Earth. Similarly, the real-time planning level (e.g., reflex response to unexpected foot-terrain collision) is not discussed as it is very specific to the final design of associated mechanisms and computing hardware.

Motion planning inputs task prescriptions and generates motion sequences that when executed perform the task as desired. The challenge for a barren terrain robot is to autonomously generate and model plan primitives considering the physics of environmental interaction, consequences of actions and the multiple, competing purposes of a mission. Motion planning generates admissible options, models them, then searches to elect those which avoid pitfalls and forward mission intentions. We foresee the existence of motion planners for AMBLER walk planning and sampling planning.

Functional specifications for the AMBLER motion planners include:

- *Implement task-level commands:* Motion planning must take occasional task prescriptions and generate motion sequences that will safely execute the task. Typical task commands might be [move North 10 m], or [find and retrieve 10 cm dia. rock]. Output sequences (trajectories) of position and/or force should include any required actuator state information (e.g., clutched, braked).
- *Incorporate relevant physics:* Task prescriptions will include constraints that are to be satisfied, such as stability margin, torque and power usage. Motion planners must therefore model all physics relevant to a proposed motion.
- *Predict confidence of intended motions:* Uncertainty from terrain and state models must be incorporated into motion planning such that planners can estimate the relative risk of an action before its implementation. Under normal operating conditions, moves with excessive risk must be rejected.
- *Plan geometrically and quasi-statically:* Assuming that the mechanism is capable of progress without resorting to dynamic moves, motion sequences should be geometry-based instead of temporal. This permits maximum flexibility for motion executors to pick and modify execution rates and command halts at any point during a motion sequence without worry of dynamic destabilization.
- *Maximize resistance to destabilization:* Planning must give the largest possible margin for execution error or unexpected event (e.g., failure of a soil slope). For instance, a self-reliant explorer that must remain upright should not rely on critical stabilizing support from an untested terrain contact. The ability to adhere to this specification is, however, quite mechanism-dependent; a rigid-chassis rolling machine has no inherent ability to choose or test its impending terrain contacts.
- *Provide acceptable limits with output motions:* Trajectories that are planned for execution must have acceptability margins, where applicable. During execution, exceeding a limit indicates a condition other than predicted by planning and may warrant halting, assessing state and replanning. Providing a limit on motor torque for a given motor trajectory is an example; if torque exceeds this value during execution of the trajectory, motion is halted.
- *Incorporate blind mode:* Motion planners must have the ability to plan task execution without input from local terrain perception. Operation in this mode assumes the use of internal position, force and proximity-type sensing. While the resulting progress would be extremely slow, it could mean the success of the Mars mission.

Configuration of a motion planner that is responsive to these functional specifications is very mechanism specific; that is, little of the motion planner can be configured without knowledge of the mechanism configuration. However, it is possible to put forth a general framework for motion planning that upholds the functional specifications.

Robot and terrain mechanics can be posed using classical techniques such as analytic energy functionals. There is significant advantage, however, to posing both of these in approximate, discrete models that are amenable to postulating and evaluating discrete states of robot-terrain interaction. Once a model of a mechanism exists, planning for such interactions becomes a cycle of enumeration, modeling and evaluation. This cycle generalizes to planning schemes for selecting route, specifying body positions, choosing foot placements and sampling in rugged terrains. The distinctions of planning for one function versus another lie only in the specifics of motion primitive, mechanism model and applicable cost penalty. For example, planning a body move for a walker might consider increments of three-dimensional motion as primitives, an energy functional as a mechanism model and energy expenditure as the cost function. Alternately, planning for a foot placement might use increments of downward motion as primitives, robot-terrain compliance as a model and robot stability as a penalty function.

The AMBLER walk_planner is now briefly overviewed. A thorough presentation and discussion of the walk_planner can be found in [5]. The planner determines foot placement locations in the terrain and calculates corresponding body and recovering leg motions to advance the vehicle safely along a commanded heading.

A significant portion of the walk_planner selects foot placements. Once foot placements are determined, the cycle of enumeration, modeling and evaluation for body motions is based on the following considerations (in approximate decreasing order of priority):

- the body must remain continuously stable *even if* any single leg fails to provide support due to a mechanical failure or failure of the underlying terrain,
- no part of the walker should collide with the terrain (only permissible terrain contacts are axial foot placements),
- body progress along the desired heading should be maximized, and
- joints torques and energy expenditure should be maintained below given thresholds.

Once foot placements and body motions have been determined, recovering leg trajectories are planned to avoid other supporting legs and terrain obstructions, and such that the foot properly contacts the terrain (axial direction only). Finally, the body motion and recovering leg trajectories along with expected peak torque values are output to a walking motion executor.

4.3. Locomotion Mechanism

The locomotion mechanism propels the robot over the terrain and implements robot posture. Functional specifications for the Mars explorer locomotion mechanism include:

- *Traversability:* The mechanism must traverse very rugged three-dimensional terrains typical of Fig-1 while avoiding self-damage and irrecoverable circumstances. The mechanism must not require an inordinate amount of terrain perception or planning to traverse such terrains at approximate speeds of 1 m/min.
- *Energy efficiency:* Total on-board power generation is expected to be less than 1 kw. As the dominant energy consumer in this minimal energy system, the locomotion mechanism must be extremely power efficient.
- *Three-dimensional motion:* The mechanism must be able to directly execute three-dimensional motions. Without this ability, a level of planning is required to decompose desired motions (e.g., move left 1 m) into a series of permissible locomotion primitives – analogous to parallel parking.
- *Predictable and assessable motion:* It must be straightforward to model the locomotion mechanism and relevant terrain geometry/mechanics from which motion can be accurately predicted. Furthermore, an estimate of subtended motion must be assessable to fairly high accuracy.
- *Incremental, reversible motion:* The locomotor must move in conservative increments such that a halt command can be immediately responded to without worry of preempting a dynamic move in progress. This ability permits reversal of any series of incremental moves, very useful for an exploratory machine that by nature moves into the unknown.

- *Reliability:* The locomotive mechanism must be mechanically simple. Furthermore, the mechanism must endure long-term environmental exposure and wear resulting from traversal of rugged terrain.
- *Payload isolation and reaction:* The locomotion mechanism must isolate most of the scientific payload from topology of the rugged terrain. The locomotor must be able to execute small incremental moves for fine positioning of science and sampling equipment. Finally, the locomotor must provide reaction for sampling equipment (e.g., core drill, soil scoop).
- *Teleoperability:* Given that occasional operator intervention is foregone, and that the Earth-Mars signal delay is significant, the locomotor should be easy to teleoperate.

Possible locomotion candidates include mechanisms that roll, walk, or combine rolling and walking for so-called hybrid locomotion. (Track laying mechanisms are grouped here with rolling mechanisms because tracks are in continuous terrain contact and are analogous to large wheels.) Any locomotion mechanism must support itself on the terrain and provide a propulsive force for motion. The fundamental differences between rolling and walking mechanisms are the means by which support and propulsion are provided. Wheeled machines have rollers in continuous support contact with the terrain and propel themselves by generating traction forces parallel to the terrain surface. Alternately, walkers suspend themselves over the terrain on discrete vertical contact points and maintain principally vertical contact forces throughout propulsion.

Wheeled Locomotion

Wheeled mechanisms are the predominant form of locomotion for most manned vehicles and unmanned robots. As a result, wheeled mechanisms and their control are well understood. Reliability, weight and performance trade-offs are also well understood. However, for the following reasons, wheeled mechanisms are not suited to the needs of an autonomous robot for exploration of the rugged Martian terrain:

- *Traversability is limited.* The need for continuous wheel contact limits the ability of a wheeled locomotor in rugged, discontinuous terrain.
- *Energy efficiency is low.* Wheeled mechanisms undergo continuous nonlinear energy losses in rugged and soft terrains due to slippage, shear and bulldozing. Furthermore, as the body rises and falls to trace terrain topology, non-conservative work is expended due to losses in the mechanism and mechanism-terrain interactions.
- *Mechanical complexity precludes three-dimensional motion.* Wheeled mechanisms that can subtend three-dimensional body motion relative to the terrain require many extra motions for omni-directional steering and suspension. The complexity of such a mechanism exceeds that of some walking mechanisms without providing similar rugged terrain performance.
- *Unable to accurately predict and assess motion.* Due to surface contact constraints, wheel compliance, three-dimensionality of rugged terrains and mechanism indeterminacy, detailed terrain-interaction models are intractable for wheel contact through difficult terrain. The consequences of a motion command in such a terrain are therefore unpredictable. Dead-reckoning, the ability of a locomotor to self-perceive incremental motion, is error-prone in mild terrains and worsens as the terrain becomes rugged, soft, discontinuous, or slippery.
- *Incremental, reversible motions are unlikely.* Most wheeled mechanisms are not quasi-static; some moves are dynamic and may not be reversible. For instance, when a wheeled machine drives over a ledge, tip-down is usually dynamic. Depending on its configuration and relative power, the machine may not be able to reverse the tip-down to retrace its path.
- *Teleoperation is easy:* As long as wheel diameter overwhelms the scale of terrain features, wheeled locomotors are inherently straightforward to teleoperate. However, as terrain features become confronting, complex moves (e.g., parallel parking type move to translate sideways) become more frequent and thus significantly complicate teleoperation.

Legged Locomotion

Walking mechanisms isolate the robot's body from the underlying terrain and propel the body independent of terrain details; thus walking mechanisms *body-terrain isolate.* Whereas wheels continuously contact the terrain, walking is a series of discrete terrain contacts. Foot contact forces for body support and propulsion are principally vertical, thus allowing more tractable models of terrain interaction than are possible for wheels.

Walking mechanisms provide the attributes required for the Mars explorer locomotion mechanism:

- *Traversability is excellent.* A walking mechanism chooses its foot placements to avoid rather than confront terrain difficulties. Furthermore, the body can be positioned relative to the ground contact points to maximize stability against tipover at all times.
- *Energy efficiency is theoretically high.* As a walker traverses rough terrain, the body is maintained at a fairly constant orientation and elevation with respect to gravity resulting in great power savings. More importantly, the power losses to terrain can be minimized by discrete foot placements.
- *Motion is three-dimensional.* Walking mechanisms achieve three-dimensional body motions regardless of underlying terrain topology.
- *Motion can be predicted and assessed.* While easier and more tractable for some walking configurations than others, walkers are able to model and monitor motions rigorously. This is mainly due to the predominantly one-dimensional nature of foot/terrain interactions.
- *Motions are incremental and reversible.* Assuming a quasi-static walker (i.e., more than 4 legs and a slow rate of progress), locomotion can be separated into discrete increments, and moves can be undone by "playing back" the incremental commands.
- *Motions are extensible.* A benefit of a walking mechanism to scientific observation and sampling is that body-mounted payloads can be positioned (hovered) over a region of interest using leg motions. Legs can also be utilized as sampling manipulators.
- *Teleoperation is difficult.* Teleoperation of a walking mechanism requires a "coordination" function that takes simple joystick inputs for Cartesian or polar referenced body moves and calculates required leg motions. Additionally, the operator must consider stability, select new foot placements, plan foot recovery trajectories and monitor body clearance.

While legged mechanisms would appear to satisfy most of the functional specifications put forth for the locomotor, existing walking mechanisms raise valid concerns regarding coordination problems, energy losses due to coordination difficulties and reliability. The fundamental coordination impediment has been the indeterminacy that attends multiple legs and motions. Coordination difficulties with these complex mechanisms lead to actuator conflict, which accounts for large energy losses. The reliability of a traditional walker is directly related to leg count and leg configuration on the body. Existing walkers are commonly immobilized after the failure of one or more legs. The AMBLER locomotor configuration should overcome these three practical concerns that elude implemented walkers to date: complexity of coordination control, resultant energy losses and redundancy for continued function after loss of some motions.

AMBLER Locomotor

Each AMBLER leg [Fig-4] consists of two rotary links (shoulder, elbow), which move in a horizontal plane and are termed the "planar" links. A prismatic vertical actuator is appended to the end of the elbow link. The AMBLER locomotion mechanism consists of six legs stacked coaxially at their shoulder joints. The actuator groups for AMBLER body support and propulsion are orthogonal; a subset of the planar joints propel the body, and the vertical actuators support and level the body over terrain. With all feet supporting load and vertical actuators locked, the body can be propelled by actuating any three of the 12 planar actuators. The remaining nine planar actuators are declutched or backdriven. The AMBLER can level itself without propelling and propel without leveling, and exhibits no power coupling between the two motions. A strength of the AMBLER configuration, therefore, is its ability to efficiently and simply enact planar moves transverse to gravity and elevation moves along the gravity vector.

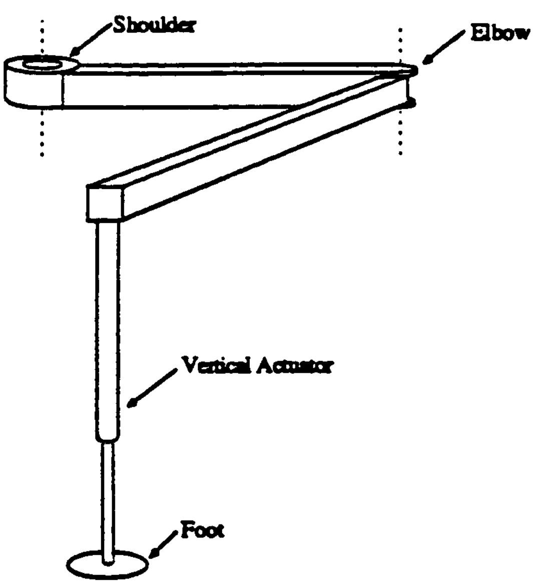


Fig-4: **AMBLER Leg**

Decoupled AMBLER body propulsion and body leveling enables decoupled formulation[2] of the planar and vertical mechanism models. The planar mechanism is determinant and can be solved very quickly in a closed-form. A one-dimensional terrain interaction model suffices for solution of the vertical foot forces. Equilibrium, stability margin, forces and energy cost are readily assessable from these mechanism and terrain models. Such accurate prediction permits torque and power minimization during propulsion. The AMBLER leg design thus enables a tractable control model and eliminates energy losses due to actuator conflict.

Internal assessment of incremental motion, termed "dead reckoning", is critical for an autonomous machine. As the AMBLER propels, all undriven planar actuators assess incremental motion. AMBLER dead-reckoning should thus be superior to other walkers, as redundant reckoning information is available from many undriven actuators.

Decoupled propulsion in the horizontal plane allows corresponding simplifications in representation and planning for the AMBLER. Since foot placements into the terrain are always vertical "punctures", a 2.5-D terrain feature model (elevation map) is appropriate for selecting foot placements. Specifically, planning a foot placement reduces to a search in the elevation map for a suitable location that will forward progress while maintaining a desired stability margin and maximum power level. Once the desired location has been selected, the foot is positioned over the terrain contact point and the vertical actuator telescopes to contact the foot on the terrain. It is our experience[6] that perception can consume 90% of the computing capacity of an autonomous mobile vehicle. The AMBLER leg design and foot placement scheme holds promise to reduce the perception bottleneck, as nearly raw elevation data suffices for selecting acceptable foot placements.

Leg Stack and Body Configuration

A fundamental locomotion configuration decision "stacked" all legs on a central shoulder axis [Fig-5] instead of 3 legs each on twin, side-by-side stacks. With the height of the planar leg section about 20 cm, a single 6 leg stack is ~1.2 m in height and the twin stack ~ 0.6 m. Motivations for the twin stack beyond height reduction included simplification of wiring and dust sealing at the shoulders of each leg.

---

[2] The decoupled formulation is based on the assumption that lateral foot slippage is prevented. The AMBLER is expected to generate relatively low foot shearing forces thus the assumption is considered valid for a majority of terrain conditions.

The best configuration proved to be the single stack, the major reasons for which follow. While the twin stack does reduce overall vehicle height, constraints on vehicle height are unsettled from the standpoint of the Mars mission. Whereas propelling the body with the single stack configuration is identical to positioning a four-bar (determinant) linkage, the twin stack configuration increases it to a five-bar linkage (the fifth link connects the centers of the two stacks). Since a five-bar linkage is indeterminate, the only way to preserve the determinacy of planar AMBLER moves would be to lock one of the joints, thus removing a link. The twin stack configuration would thus place the added burden on walk planning to lock certain joints for determinant planar body motion. Additionally, some means to lock or brake the joints would have to be mechanically incorporated.

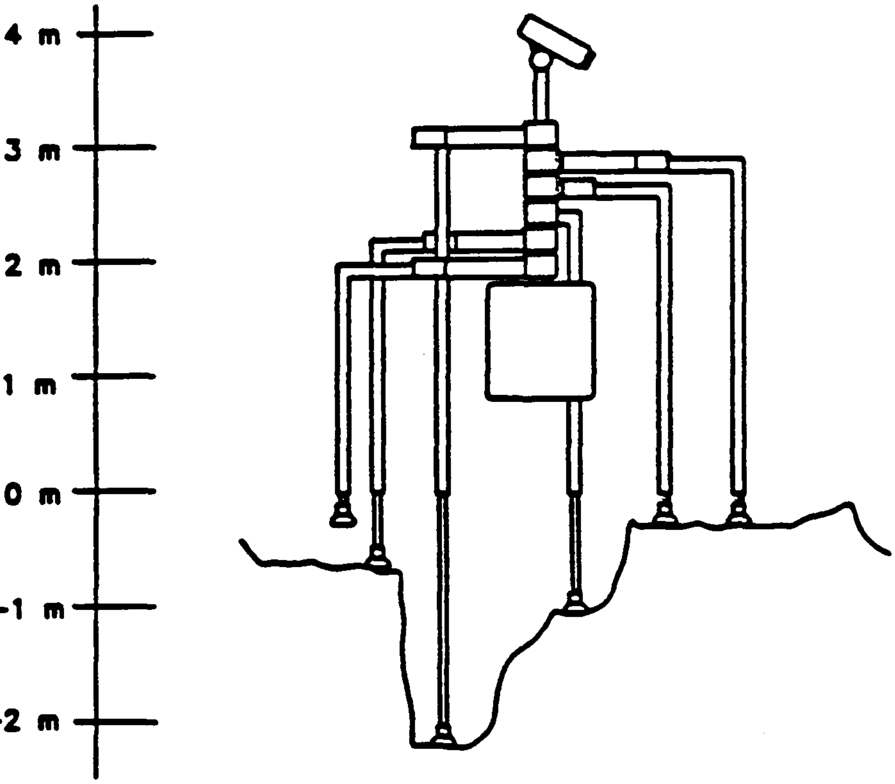


Fig-5: **AMBLER Leg Stack**

A further benefit of the single stack leg configuration is redundancy – the ability to operate any leg in any body sector dramatically improves the reliability of the AMBLER over conventional walking mechanisms. Any functional leg can reposition itself to substitute for any failed leg. Three legs would have to fail to cause immobilization. Resorting to the twin-stack would roughly halve the redundancy potential of the AMBLER as legs would only be able to functionally replace other legs in the same stack.

With the single stack configuration, the vertical actuator on the uppermost leg is roughly 1 m longer than the vertical actuator of the lowest leg. This extra length is utilized by adding stroke to the upper legs. Leg stroke varies from ~1 m on the shortest leg to ~2 m on the tallest. The added leg strokes provide more planning options for level body motion during steep hill climbing (i.e., trail legs with longer strokes) and traversal of areas with deep holes or ruts (i.e., place legs with longer strokes in holes – shown in Fig-5).

A unique result of the stacked leg configuration [Fig-5] is that gaits where rear legs recover *past* forward supporting legs are ideally possible. These overlapping gaits mean fewer foot placements with advantages of reduced demand on perception and planning, and significant energy savings due to reduction of the number of foot-terrain interactions. An example is shown in Figure-6; as the walker advances, the recovering leg (shown in bold with dashed trajectory) overlaps the two right side supporting legs. Depending on a leg's location on the central stack, some weaving around supporting legs may be necessary for it to recover past forward legs.

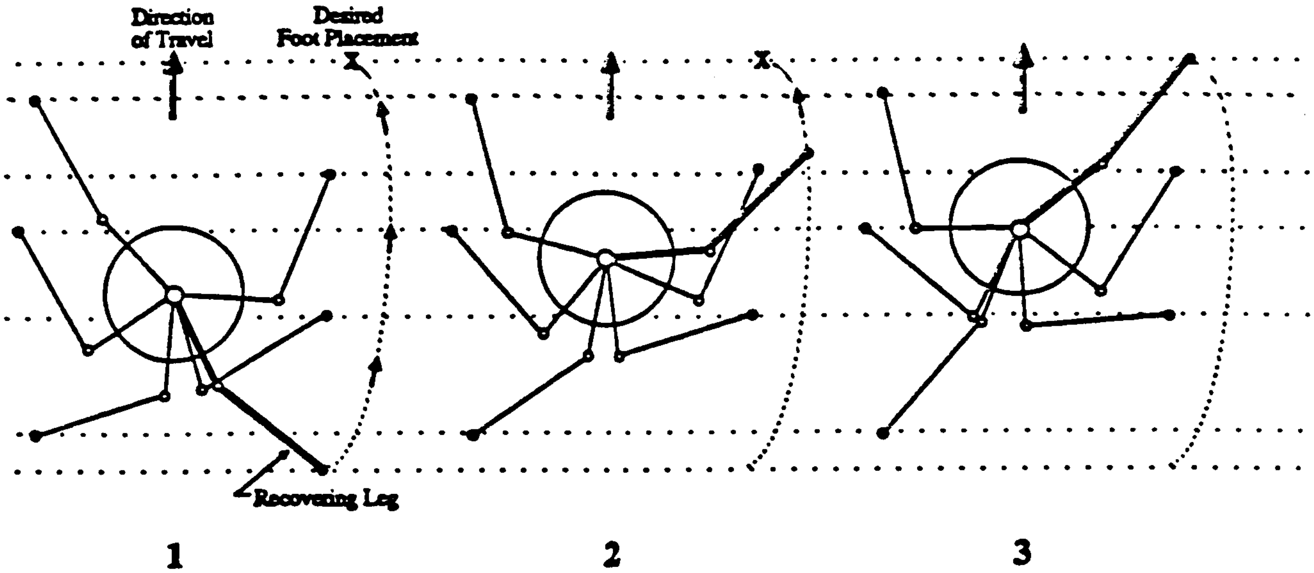


Fig-6: Overlapping Gait (plan view)

The body, containing science and sampling equipment, computing, power and communication systems, is suspended below the leg stack. Placing the body below the leg stack has multiple benefits: The vehicle center of gravity is significantly lowered. Scientific sensors and sampling equipment mounted under the body have unobstructed downward terrain views. The body can be lowered to the ground (body grounding) to enhance vehicle stability and rigidity for high reaction sampling tasks such as deep coring.

Leg Drivetrain Configuration

It is possible to degenerate AMBLER walking to an infinite sequence of decoupled primitive movements. Each cycle consists of a foot lift, shoulder rotate, elbow rotate, foot lower and lock and body propulsion. Body propulsion can be further decoupled to a sequential actuation/braking of any three planar motions. The mechanical implication is that the entire 18-DOF AMBLER could be operated from a single gearmotor with a series of clutches, brakes and drivetrain components to direct torque to any of the 18 joints. The other extreme is a gearmotor at each joint. Any number of gearmotors between 1-18 could be configured to drive the AMBLER locomotor.

The most significant factors influencing this decision concerned overall reliability, power efficiency, and implications to walk planning. Any configuration with less than 18 gearmotors places additional burden on walk planning to select proper clutch/brake states to properly direct torque from shared gearmotors. Configurations with gearmotors housed within the stationary central axis and drivetrains emplaced through the shoulders were ruled out due to the mechanical complexity required to transfer large torques while still preserving 360° shoulder motion. A mechanically straightforward solution configured a single shared gearmotor in one of the planar links of each leg, with drivetrains and clutching to direct torque to any one of the three leg actuators. However, the configuration was deemed unacceptable because coordinated joint schemes enabling a recovering leg tip to trace a general trajectory in space were not possible. Drivetrain complexity, significant power losses and reduced system reliability of configurations with shared gearmotors led to a single gearmotor configured at each leg joint. This configuration also permits complete motion and control flexibility for walk planning and control schemes.

During body propulsion only a subset of the planar joints are driven; the balance are non-powered. As a non-powered joint is moved, its gearmotor is backdriven. Alternately, a clutch can be added to the output of the gearmotor to decouple the gearmotor from the joint during the non-powered state. Backdriving eliminates the need for a clutch at each planar joint, but induces higher torques on the propelling gearmotors. Backdriving also eliminates the control task, albeit fairly simple, of engaging/disengaging clutches to select the desired propulsion gearmotors. Compounding the decision was that clutch power draw and severity of gearmotor backdriving torque cannot be accurately predicted. For these reasons, the mechanically simpler solution of backdriving was chosen.

5. Conclusion

The Mars Rover/Sample Return mission aspires to send an unmanned robot to the distant planet for a several year mission during which it will traverse hundreds of kilometers and conduct a variety of scientific activities. The AMBLER is a robot configuration to meet the challenge. The scope of the AMBLER configuration includes hardware, software and processes for perception, planning and action. The configuration is the basis for continuing detailed design and production of the prototype walker.

Local terrain perception, motion planning and locomotion subsystems combine to form the core of the AMBLER robot. The synergy between these subsystems is essential to the success of the prototype system. Many AMBLER subsystems were not discussed in detail, including intermediate and global perception, task planning, motion control, science/sampling, computing hardware, health monitoring, communication and infrastructure. Configuration of each of these subsystems is also critical to the functioning AMBLER, and their configuration is completed, though the scope of this paper precluded further detailing.

A new generation of mobile, perceptive robots is needed to explore and work in natural terrains on behalf of man. The challenge motivates a class of robot with unprecedented ability for autonomous operations in environments characterized by rugged terrain, soft soils and harsh meteorological conditions, such as the surface of Mars. Beyond an immediate relevance to planetary exploration, these robots will evolve for duty in Earth applications like military reconnaissance. Successors to these exploratory robots will excavate, mine, and till the barren terrains of Earth and the planets. Exploration is invaluable as a precursor to more aggressive robot functions that forcefully modify their environments, and in the case of planetary exploration, as precursors to man's presence.

6. References

[1] Mars Study Team, A Preliminary Study of Mars Rover/Sample Return Missions, Solar System Exploration Division, NASA Headquarters, January 1987.

[2] M. Hebert, and T. Kanade, "3-D Vision for Outdoor Navigation by an Autonomous Vehicle," in Proc. of Image Understanding Workshop, Morgan Kaufmann Publishers, Cambridge, Massachusetts, April, 1988, pp. 593-601.

[3] W.K. Stewart, Ph.D. Thesis, Massachusetts Institute of Technology/Woods Hole Oceanographic Institution, December 1987.

[4] Results from I. Kweon, M. Hebert, and T. Kanade, "Rough Terrain Perception," in Proc. of Society of Photo-Optical Instrumentation Engineers, in press.

[5] S. Mahalingam, M.S. Thesis, "Walk Planning for a Rugged Terrain Robot," University of North Carolina at Charlotte, August 1988.

[6] S. Shafer, A. Stentz, and C. Thorpe, "An Architecture for Sensor Fusion in a Mobile Robot," in Proc. of the IEEE International Conference on Robotics and Automation, April 1986.

A Case Study in Robot Exploration

Long-Ji Lin, Tom M. Mitchell
Andrew Philips, Reid Simmons

CMU-RI-TR-89-1

Computer Science Department
and
The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

January 1989

© 1989 Carnegie Mellon University

This research was supported by NASA under Contract NAGW-1175.

Abstract

This paper reports on a case study in autonomous robot exploration. In particular, we describe a working mobile manipulator robot that explores our laboratory to identify and collect cups. This system is a first step toward our research goal of developing an architecture for robust robot planning, control, adaptation, error monitoring, error recovery, and interaction with users. We describe the current system, lessons learned from our earlier failures, organizing principles employed in the current system, and limits to the current approach. We also discuss the implications of this work for a more robust robot control architecture which is presently under development.

1. Introduction

We report on a case study in autonomous robot exploration. In particular, we describe a working mobile manipulator robot that explores our laboratory in search of cups. This system is a first step toward our research goal of developing a robust architecture for robot exploration tasks, covering planning, control, error monitoring and recovery, adaptation, and communication with users. It is also intended as a testbed for better understanding the task of semi-autonomous robot exploration and sample collection. The robot exploration task is of specific interest to us, given a related effort to develop a prototype robot to explore the surface of Mars to collect geological samples [1]. This testbed is thus intended both to help explore characteristics of the Mars Rover task, and as a general carrier for a broad range of research on autonomous intelligent robots.

The robot exploration task considered here is one in which a mobile robot with an attached manipulator explores an area using vision and sonar sensors in order to locate and identify cups. When a cup-like object is located, the robot navigates to it and uses more detailed sensing to determine whether it is truly a cup, and if so what type. It then picks up the cup, travels to a box, deposits the cup, and looks for additional cups to collect.

This task raises a number of general issues that must be addressed for exploration tasks, as well as specific issues that must be addressed in the Mars Rover scenario. These include:

- Path planning and navigation
- Observing and identifying encountered objects
- Integrating locomotion with manipulation and perception.
- Maintaining background goals (e.g., battery charge level) while pursuing the current goal (e.g., pick up the object).
- Detecting errors (e.g., the cup was not grasped correctly) and recovering from them.
- Communicating and collaborating with a remote human for guidance in dealing with difficult tasks.

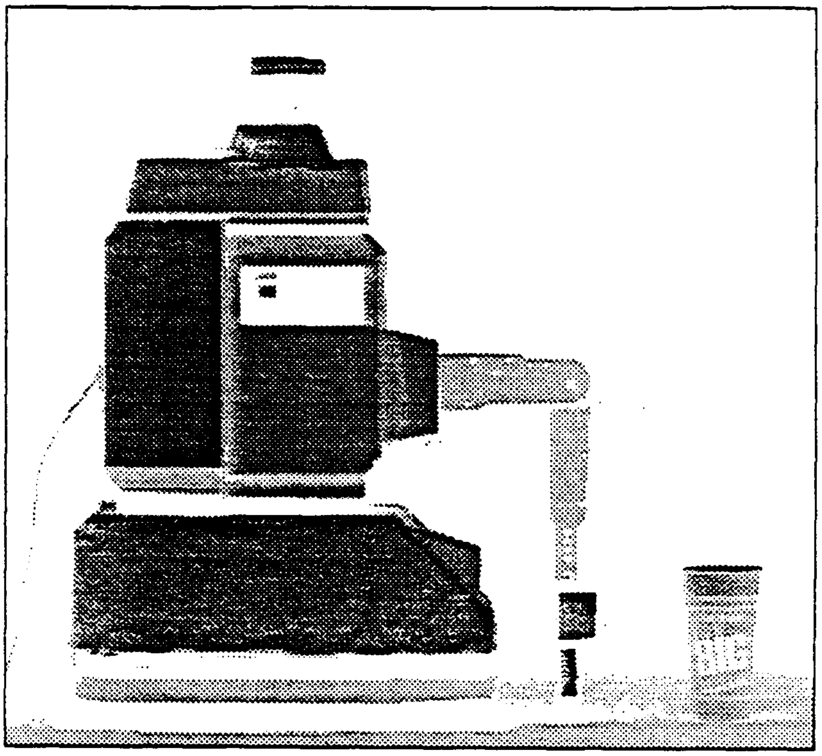
Our present system deals well with some of the above issues, and poorly with others. The implemented system autonomously locates cups, navigates to them, picks them up, and deposits them in a bin. However, it does not presently manage multiple goals, deal well with unexpected contingencies, or collaborate with humans.

This paper describes the current system, lessons learned from our earlier failures, organizing principles employed in the current system, and limits to the current approach. We also discuss what we have learned from this work regarding specific problems that must be addressed by the architecture currently under design. Section 2 describes in greater detail the hardware setup, task, and approach taken for this exploration task. Section 3 characterizes the performance of the system, including interesting failures which it has exhibited. Finally, section 4 characterizes lessons learned from this case study, and implications for the design of more robust architectures for robot planning and control.

2. System Description

2.1. The Robot Testbed

Figure 2-1: A Modified Hero 2000 Robot



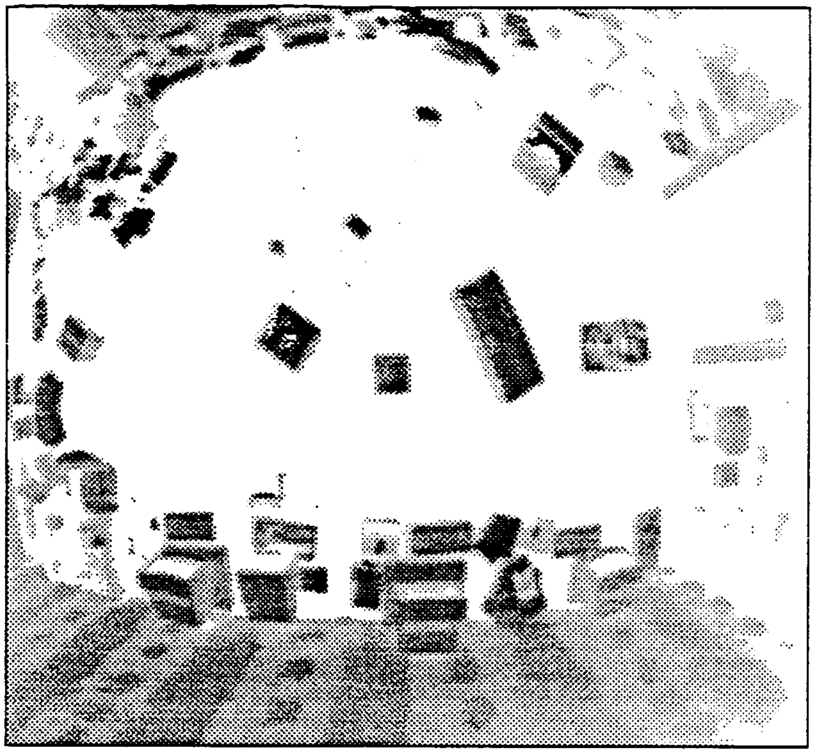
The robot used is a commercially available wheeled robot with arm (the Heath/Zenith Hero 2000), as shown in figure 2-1. The robot is located in a laboratory in which a ceiling-mounted black and white television camera is able to view the entire room through a fisheye lens. Figure 2-2 provides a view of the room as seen through this ceiling camera. The Heath robot comes with two standard sonar sensors: a rotating sonar on the top of the robot which completes a 360 degree sweep in a little over a second, plus a second sonar fixed to the base of the robot and pointing forward. In addition, we have added a third sonar to the hand of the robot. Since this third sonar is located on the hand of the robot, it can be repositioned relative to the robot body. We have found that this capability is important for smooth integration of manipulation and locomotion operations. The cost of this setup is approximately $15,000 (in addition to the cost of the Sun workstation).[1]

The robot contains an onboard microcomputer (based on an Intel 8086) which executes all primitive motion and sensing commands. It communicates with a Sun workstation running C and Lisp programs. Communication between the robot and Sun may be via either a radio link at 600 baud, or an RS232 cable

---

[1] We are considering making the plans and software for this setup available to other universities and research laboratories. Interested parties should contact the authors.

Figure 2-2: Overhead View of Laboratory as Seen by Robot



at 9600 baud. In practice, we have found the 600 baud radio link constitutes a communications bottleneck, and therefore frequently utilize the more awkward but faster RS232 tethered connection. Table 2-1 summarizes the sensor, effector, and computational characteristics of the robot testbed.

2.2. The Task and Approach

As stated earlier, the robot task is to collect cups into a container in the corner of the lab. The top-level procedure used by the system to accomplish this task is described in table 2-2. Below we discuss in additional detail each of the steps in this high-level plan.

Locate robot, potential cups, and obstacles. The system begins by examining the visual image from the ceiling camera to locate regions that correspond to potential cups, the robot, and other obstacles. The image is thresholded and regions extracted by the Phoenix program [5]. The robot region is located based on searching a window within the visual field, centered around the current expected location of the robot. Within this window, the robot region is identified based on a simple model of the properties of its region in the visual field. Potential cup regions are identified by searching the entire visual field for regions whose size and shape match those of cups. Since the resolution of this image is fairly coarse (approximately 1 inch per pixel), and since a simple thresholded black and white image is used, it is possible for the system to identify non-cup regions (e.g., sneakers worn by lab residents) as potential cups. In figure 2-2, it is possible to see several cup-sized regions in the image. The robot will navigate to each of these regions, using its sonar to explore each in turn. Those which it eventually determines are not cups are remembered as such, in order to avoid examining them repeatedly.

Table 2-1: Robot Testbed Summary

Effectors:
Heath/Zenith Mobile Robot with Arm
Torso rotates relative to base
Arm mounted on torso
Zero degree turning radius
Locomotion precision in laboratory environment returns robot to within a few inches of initial position when commanded to navigate a 10 foot square

Sensors:
Overhead (ceiling-mounted) camera
Obtains 2D visual regions across entire lab
Approximately 1 inch resolution

Forward-pointing sonar on robot base
(all sonars have range 4-127 inches, distance resolution .5 inch, uncertainty cone 15 degrees)

Rotating sonar on robot head
360 degree sweep in 15 degree increments in 2 seconds

Movable sonar fixed to robot hand can be repositioned relative to body

Battery charge level sensor

Rotating light intensity sensor on robot head

Computation:
Speech synthesizer and microprocessor onboard
Radio link (600 baud) or RS232 cable (9600 baud) to Sun workstation
A MATROX frame-grabber board on the Sun is used to digitize images
Generalized Image Library is used to create, maintain, and access image files [3].

Navigate to vicinity of target object. Once a target object is located, a path is planned from the current robot position to the vicinity of the object. A path consists of a sequence of straight line segments and zero-radius turns. The path planning algorithm models the room as a grid of robot-diameter-sized grid elements, and utilizes Dijkstra's shortest path algorithm to compute an initial path. In choosing this path, the system takes into account (1) proximity of obstacle regions, (2) total path distance, and (3) number of vertices in the path. It then optimizes the path by adjusting each vertex in the initial path, using local information to minimize the cost of the path segments on both sides of that vertex. The basic idea behind our algorithm is grid search and path relaxation proposed in [8]. Figure 2-3 shows an interpreted version of the image from figure 2-2, along with a path planned by the system to reach a potential cup region and a uncertainty cone(see below). Here, the brightened line shows the final computed path, while the dimmer line is the original path before optimization.

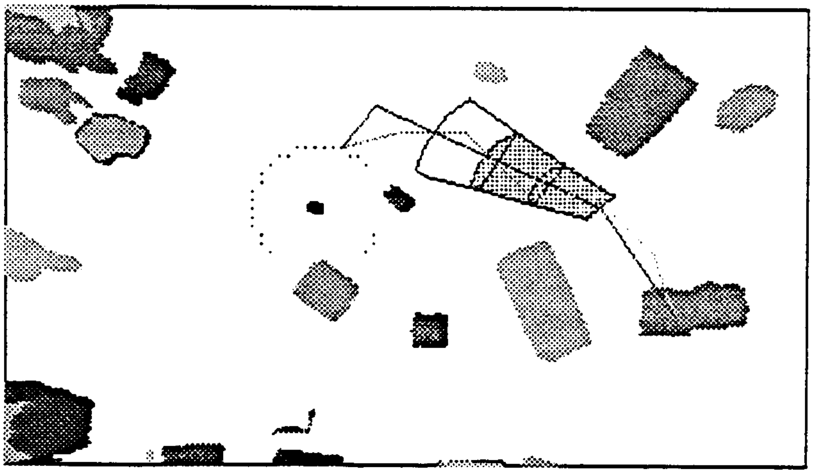
Once a path is completely planned, the robot begins to follow it. At certain intervals the robot stops, uses the ceiling camera to determine its progress, and updates its path accordingly. The system utilizes an explicit model of sensor and control uncertainty to determine how far the robot may safely proceed

```
Do until no potential target objects remain:

    Locate robot, potential target objects, and obstacles within visual field
        Find visual regions
        Identify those with appropriate features
    Navigate to vicinity of target
        Plan obstacle-free path to vicinity
        Move along path, monitoring with vision
    Approach and identify target
        Use sonar to locate nearest object in appropriate direction
        Servo using sonar until target centered at 0 degrees, 6.5 inches ahead
        Classify object as non-cup or specific type of cup
    Grasp cup, based on identified cup type
        Make final approach to grasping position
        Move arm and gripper to grasp cup
        Use top sonar to determine whether arm successfully grasped object
    Configure arm and body for safe travel
    Navigate to container
    Orient to center container in front of robot
    Deposit cup in container
```

Table 2-2: Top-Level Cup Exploration Procedure

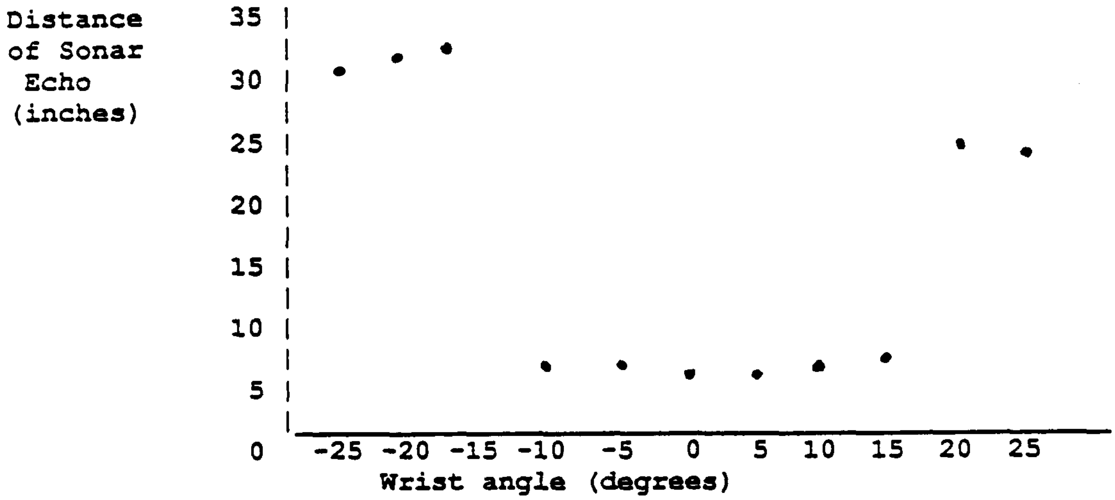
Figure 2-3: Interpreted Overhead View With Planned Path and Uncertainty Cone



along its path before a new visual check is required. A covariance matrix representation [7] of uncertainty is used. Robot location and orientation are calculated by merging information from both dead reckoning and vision. The system introduces a new sensing operation when uncertainties in sensing and control have grown to the extent that either (1) collisions with obstacles are possible, (2) the uncertainty modeler is unable to model uncertainty accurately, or (3) the visual recognition routines which utilize strong expectations about robot location do not have strong enough expectations to operate reliably.

By modifying an old path, the path planner can efficiently adapt to small environmental changes, such as the robot wandering slightly off the planned path, new obstacles appearing, and old ones disappearing.

Figure 2-4: Sonar Data Obtained by Wrist Sonar Observing a Cup



The data in angle-distance pairs is:

(-25 31) (-20 32) (-15 32.5) (-10 7) (-5 7)
(0 6.5) (5 6.5) (10 7) (15 7.5) (20 24.5) (25 24).

The data was taken while two boxes were placed on the background of the cup.

Approach and identify target. Navigation under the direction of vision is able to place the robot within several inches of its desired location. In order to successfully grasp an object, however, the relative position of the robot and object must be controlled with significantly greater precision (on the order of an inch or less). Thus, once the robot reaches the vicinity of the target object, it utilizes its sonar to locate itself more precisely relative to the target object and to classify it. Figure 2-1 shows the pose which the robot assumes in order to utilize its wrist sonar to detect the location and dimensions of the object. The wrist is rotated from side to side in order to sweep directly ahead and detect the object. This sweep provides a one-dimensional horizontal array of sonar data giving distance readings as a function of wrist angle. Figure 2-4 shows a typical set of data obtained by the hand sonar when observing a cup in this fashion. Simple thresholding, edge finding, and region finding routines are then used to process this one-dimensional array in order to locate the object in the sonar field of view.

Once the object is located in the sonar field, its distance and orientation are used to compute robot locomotion commands to bring the object to 0 degrees (plus or minus 2 degrees) and 6.5 inches (plus or minus .5 inch) in front of the wrist sonar. To overcome sensing and control errors, this procedure is repeated after the locomotion commands are executed until the sonar detects the object at the desired position. Typically, this servo loop requires from 1 to 3 cycles before convergence. Once in position, the object width and height are measured (in degrees of wrist motion) to identify the object as either (a) an upright standard-sized styrofoam coffee cup, (b) an upright Roy-Rogers Big Chiller cup, or (c) neither.

Grasp object. If the object is identified as one of the two known types of cups, then it is grasped by a procedure specific to that object. The grasping operation itself does not use sensory feedback to guide the hand--all the sensing work is performed during the precise positioning of the robot during its approach to the object. With this reliability in positioning, it is fairly rare for the grasping operation to fail. Note that the lack of a need for sensing here is due in part to the fact that the object shape and dimensions are know a priori, and to the predictability of the physics of interaction between the gripper and object. If one were to attempt picking up a rock of unknown size and shape half buried in the sand, significant sensing would most likely be essential to monitor the details of the manipulation operation.

Once the grasp action is completed, the robot raises its hand so that the object which it is (presumably) holding may be detected by the head sonar. This check allows the system to verify the success of the grasping operation. If it instead detects failure, then the system labels the corresponding visual region as a "fools-cup" and subsequently avoids it. No attempt is presently made to replan or recover from manipulation errors, though this is a topic we intend to pursue in the future.

Navigate to container and deposit cup. Once a cup has been successfully obtained, it is tucked in to protect the arm during travel, and a path is planned to the container. The cup is then deposited, and once again the system looks for a new cup to collect.

3. System Performance

The system described above is fairly successful at locating and collecting cups. We estimate that it succeeds approximately 80-90% of the time in finding, retrieving, and depositing cups that are placed on the floor away from the perimeter of the room in unobstructed view of the camera. It typically requires on the order of 5 minutes to locate a candidate cup, navigate to it, pick it up, and drop it off in the container (when communicating via the 9600 baud tether). Approximately half of this time is spent navigating to the cup and later to the container to drop it off. The other half is spent near the cup, refining the relative position of the robot and cup, identifying the object, and grasping it. The overall time increases by a factor of three when using the 600 baud radio link, indicating that when the radio link is used the system bottleneck is the low baud rate communication link which must pass commands from the Sun to the robot, and sensor data from the robot back to the Sun.

Since many of the most interesting lessons we have learned arise from observing failures of the system, we summarize several of these encountered failures in table 3-1. The point to notice about these failures is that they typically arise either because of lack of appropriate sensing (e.g., the collision of the hand against the table when picking up a cup under the table edge), outright errors in sensing (e.g., when the short cup is not found by the sonar), or because of our lack of imagination in anticipating the many possible interactions between subparts of the procedure (e.g., that after picking up the cup, the robot would refuse to move because the vision system saw the cup held in front of the robot as an obstacle!).

Perhaps some of these errors could have been avoided had we originally included more sensing on the robot's part or more imagination on our own. However, the nature of unengineered environments is that they provide a continual source of novel and unanticipated contexts, interactions and errors (e.g., a cup found beneath the edge of the table, a second cup positioned unfortunately near the target cup so that it is run over). It seems unlikely that one can expect to anticipate all possible interactions and novel situations in advance.

Given that such unanticipated events are bound to occur, and given that the system cannot afford to sense everything that goes on in the environment, an important question is exactly what needs to be sensed at a minimum to assure basic survival of the robot, and what kinds of sensors and sensor

Table 3-1: Typical System Failures and Causes or Repairs

Vision system fails to find robot.

- This can happen when the robot region overlaps another visual region. This is usually not fatal, since the system then uses the expected robot location (based on dead reckoning) in place of observed location, and proceeds.

Misses container when it deposits cup.

- This can occur due to sensing error in the visual determination of the robot orientation relative to the container. This could be overcome by more tedious servoing with the sonar to center the robot in front of container to some desired tolerance.

Robot fails to dock successfully with battery charger.

- This is due to the fact that we initially underestimated the tolerance of the docking element to error in the robot position, and overestimated the sensor error involved in using the sonar to position the robot relative to the docking element. As a result, the system refuses to dock because it believes it is not positioned precisely enough relative to the docking element, *despite the fact that it is!* This failure is interesting in that it is a direct consequence of the difficulty of estimating sensor and control errors in advance.

Runs over other cups when trying to grasp one of them

- This is because approach and grasp routines do not watch out for obstacles.

Execution monitoring causes failure if person walks through field, and is seen as an obstacle

- This is because system does not distinguish moving from non-moving objects.

Finds non-cup objects (e.g., sneakers) which appear visually to be cups.

- These are generally identified as non-cups upon closer examination. But they can result in considerable wasted time.

Time was wasted conducting sonar sweeps at needlessly detailed resolution while positioning robot relative to cup.

- This is due to the fact that we could not accurately know in advance how fine-grained the sonar sensing should be (i.e., collect a reading every 2, 5, or 10 degrees). Once we experimented and determined that we had chosen an overly conservative value, we decreased the resolution to increase efficiency without increasing the error rate.

Arm collides with table after picking up cup that is under edge of table.

- This occurs when the robot raises its hand to use the head sonar to determine whether it has successfully grasped the cup. Due to failure to check hand trajectory for collisions.

Robot unable to navigate to container, because grasped cup is now seen by vision system as an obstacle in front of robot(!)

- This was repaired by having the robot hold the cup *behind* itself. Vision still sees the cup as an obstacle, but now it is behind the robot.

placements simplify the processing of this sensor data. As an example of a reasonable sensing strategy, consider that by implementing arm motions so that the (movable) wrist sonar is pointed in the direction of the arm sweep, it is possible to use this sonar sensor as a proximity sensor to detect unanticipated collisions before they occur. This strategy would allow the system to avoid damaging itself even when unanticipated situations arise, such as raising the cup from beneath the table. Once this sensed condition were detected, it could be used as a trigger to attempt to explain and recover from the failure.

4. Lessons from The Case Study

This section summarizes organizing characteristics of the current system, discusses the impact of uncertainties on the task, and suggests capabilities that we intend to incorporate in future extensions to the current system.

4.1. Characteristics of the Prototype System

- **Few general-purpose approaches were needed.** Although the general problem of planning arm trajectories and grasping motions is very difficult, we found little need for such methods. Instead, we defined a simple, fixed, blind grasping routine, determined the context in which it would succeed (i.e., the relative position and orientation of the cup and the tolerance to error in this relative position), and then designed the remainder of the system to assure that the robot would position itself so that this specialized routine would succeed. Thus, the system gets away with simple, specialized grasping at the cost of stronger demands on the routines that must position the robot relative to the cup. A similar situation holds for the problem of object identification. Classifying object identity from an arbitrary distance and vantage point is a computationally demanding task, which is avoided in this case by servoing to a known vantage point before attempting to identify the object[2]. The acceptability of such specialized procedures for grasping and object identification suggests that solutions to general problems can sometimes be found by embedding specialized methods inside larger procedures that assure these procedures are only invoked within specialized contexts. This system organization involving collections of coupled, but specialized, routines is similar to that advocated in [2]. The one major case in which general purpose planning is used in the system is in the path planning component. We see no way to avoid the need for such general-purpose solutions in this case.

- **Explicit reasoning about sensor and control uncertainty allows intelligent utilization of expensive sensing operations.** The first implementation of the system monitored the robot navigation by employing a visual check at each vertex of the robot's path. This was subsequently replaced by a strategy that selects appropriate sensing operations based on a model of the vision sensing and robot motion uncertainties as well as the positions of obstacles. This shift resulted in *both* a significant speedup/reduction in the number of vision operations typically performed, *and* an increase in reliability of navigation in cluttered environments.

- **Both low-level and high-level sensor features used in decision making.** The sensor data is generally interpreted in terms of features at differing levels of abstraction. For example, a sonar data sweep gives rise to a one-dimensional array of distance versus angle readings. This array is interpreted to find progressively higher level features such as sonar edges, regions, region widths, and object identities. We found it useful for the decision-making procedures of the robot to utilize all of these levels of interpreted data in various contexts. For example, raw sonar readings are used to determine whether the cup is in the robot's hand, whereas sonar edges are used to decide on the object height, and region widths are used to determine the object diameter. This suggests that it will be important for the perception module of our new architecture to allow access to sensor data at multiple levels of abstraction.

---

[2] Note the fact that the cup is a cylindrical object allows cleanly separating object identification from positioning the robot at a known vantage point relative to the object. It would be interesting to extend this approach to objects that lack this cylindrical symmetry.

- **Multiple coordinate frames found useful.** We also found it useful for the system to reason in differing coordinate frames. The world-centered coordinate frame is used for path-planning and navigation tracking, whereas a wrist-oriented coordinate frame is used to describe the expected dimensions of the known types of cups (since this is the coordinate frame in which the raw sensor data is produced). We also found it easiest to use the wrist coordinate frame to describe the desired and observed position of the cup relative to the robot. Converting to the world-coordinate frame in this case introduces needless computation and rounding errors (though it is possible that doing so would make it easier to avoid obstacles whose positions are described in the world coordinate frame).
- **Communications bottleneck indicates that computational complexity is relatively low.** The fact that the 600 baud radio link causes a very significant slowdown in the overall system is an indication that the processing demands of this task are relatively small compared to communication demands. Of course it is not clear that this would continue to be the case in less structured environments, or as the system is scaled up to handle more sensors, or to respond to dangerous situations in real time.

4.2. Pervasive Uncertainty

The robot faces many types of uncertainty. It lacks a perfect description of its world, because its sensors cannot completely observe the world. In addition, it lacks a perfect characterization of the effects of its actions, so that even if it had a perfect characterization of its world it (and we) would have difficulty constructing perfect plans in advance of executing them. These are commonly cited difficulties of real-world robotics problems.

One type of uncertainty that has been especially important in the development of this system is our own uncertainty about the sensor and control errors of the robot. For example, when developing the routine to position the robot in front of the cup, we did not know what resolution to use for the sonar sweep (i.e., should the robot scan from -45 to 45 degrees in 1 degree increments, or something else). We also did not know how precisely the robot would have to be positioned relative to the cup (0 degrees and 6 inches, plus or minus *what* error tolerance?). In fact, we simply picked numbers for these parameters, and then tested the system. If it failed to successfully grasp the cup, we increased the sensor resolution or the positioning tolerance. If it succeeded but operated too slowly, then we decreased these parameters. The point is that correct values for these parameters are impossible to derive in advance, unless one knows in detail the sonar reflectance properties of the object, the spread in the sonar signal as it travels, the tolerance of the gripping action to errors in relative position, etc. We did not know these facts, but found it fairly simple to guess some initial parameter values and then increase or decrease as needed.

This has significant implications for the feasibility of automatic planning by the robot to deal with new situations (consider something as simple as planning to pick up a new type of cup). We believe it may be easier for such automatic planning to proceed by selecting and then adapting parameter values through experience, just as we found ourselves doing, rather than attempting to plan correctly all parameter values based on a detailed analysis of the physics and models of sensor and control errors (as suggested, for example, in [4]). We intend to explore this type of robot learning in the future.

4.3. Target Capabilities for the Task Control Architecture

We are presently reimplementing an extended version of the prototype system within a more principled architecture that is intended to increase the robustness of the system [6]. In particular, we intend to this architecture to provide new capabilities including:

- **Reacting to a changing world.** If the system is attempting to reach a cup and the cup tips over, or someone picks it up, or a new obstacle appears in its path, the robot should react appropriately to these changes in its world. To do so requires at a minimum the sensing capability and focus of sensor attention to detect such changes. But it also requires determining an appropriate response in an appropriate time frame, while gracefully discontinuing the current activity of the robot. Our new architecture is intended to support such reactive abilities by maintaining dependencies between sensed data and current goals and subgoals. Such dependencies will be used to determine which, if any, current goals or beliefs should be revised in the face of changing sensor data.

- **Supporting multiple goals.** The current system has no explicit goals, though implicitly its procedures cause it to appear to exhibit goal-directed behavior. A realistic system should have multiple explicit goals (e.g., "maintain the battery charge", "obtain cups", "avoid obstacles"). We intend for our architecture to support such multiple goals, and to be able to switch among them as appropriate. For instance, if the robot is approaching a cup and finds that its battery charge is becoming dangerously low, it should suspend its attempts to achieve the "obtain cup" goal in order to attend to the higher priority "recharge battery" goal, and then later resume the interrupted activity.

- **Temporarily overriding or undoing achieved goals.** Once the system has multiple goals, then subtle interactions can occur. For example, if the robot is carrying the cup to the container and encounters an impassable field of obstacles, then it might need to put the cup down and use its hand to clear its path of obstacles. Afterwards, it should pick up the cup and continue to pursue its original goal. This type of overriding and undoing a partially achieved goal (putting down the cup which has already been successfully grasped) and later resuming, is typical of the kind of goal interactions we believe our architecture must support.

- **Detecting and recovering from errors.** The present system is able to detect some types of errors (e.g., to determine that it failed to grasp the cup). We intend for our architecture to support more complete error detection as well as reasoning about how to recover from certain types of errors. For example, if it is determined that the system failed to pick up the cup, the system should attempt to characterize why (e.g., it was not a cup, but only a round piece of paper on the floor; or the cup was tipped over during the grasping operation), and to replan accordingly. General error detection and recovery is extremely difficult, but we believe that all dangerous errors must at least be detected and dealt with to assure the survival goal of the robot is maintained. Beyond that, we also intend to explore recovering from certain errors in a fashion that allows the original goal to be effectively achieved.

- **Collaborating with remote human advisor.** We desire for our system to communicate with a remote human advisor/commander in order to obtain new commands and to obtain advice about how to deal with difficult situations that arise in pursuing its goals. This is an especially important issue in the context of the Mars Rover project, in which such collaboration must occur under the constraint of large time delays. Here, the usual methods of human teleoperation do not work well. Instead, the robot must play a much greater role in deciding when intervention is needed and what information to send to the human to allow him/her to help make the decision. In the context of the current testbed, we intend to study such issues by allowing the robot to communicate with a person in another room. For example, if the robot finds that it has failed to place the cup correctly in the container, then it may decide (a) to try again, (b) to ask for assistance and send appropriate information regarding the current situation and plausible cause of the error, or (c) to do both in parallel.

5. Acknowledgements

We are grateful to John Allen, who helped set up the original robot equipment and helped develop the interface between the Sun and robot. We thank Takeo Kanade and his group for providing the Generalized Imaging Library routines which provide the low-level vision processing in this system. Jim Moody has been a great help in getting the vision hardware and software set up correctly. Similarly, Steve Shafer provided expertise and assistance in helping select appropriate vision hardware and software. This research has been supported by NASA under Contract NAGW-1175.

References

[1] Bares, J., et al.
An Autonomous Rover for Exploring Mars.
IEEE Computer Magazine , .
Special Issue on Autonomous Intelligent Machines. Submitted September, 1988.

[2] Brooks, R.A.
A Robust Layered Control System for a Mobile Robot.
IEEE Journal of Robotics and Automation 2(1), March, 1986.

[3] Hamey, L., Printz, H., Reece, D., and Shafer, S.A.
A Programmer's Guide to the Generalized Image Library
Carnegie Mellon University, 1987.
CMU IUS document.

[4] Lozano-Perez, T., Mason, M.T., Taylor, R.H.
Automatic Synthesis of Fine-Motion Strategies for Robots.
International Journal of Robotics Research 3(1):3-24, 1984.

[5] Shafer, S.A., and Kanade, T.
Recursive Region Segmentation by Analysis of Histograms.
In *Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing*, pages 1166-1171. IEEE, May, 1982.

[6] Simmons, R., and Mitchell, T.M.
A Task Control Architecture for Mobile Robots.
Technical Report, , 1989.
Submitted to AAAI Symposium on Robot Navigation.

[7] Smith, R.C., and Cheeseman, P.
On the Representation and Estimation of Spatial Uncertainty.
In *The International Journal of Robotics Research*, pages 56-68. 1986.

[8] Thorpe, C.E.
FIDO: Vision and Navigation for a Robot Rover.
Technical Report, CMU-CS-84-168, Carnegie Mellon University, 1984.

Perception for Rugged Terrain [1]

In So Kweon, Martial Hebert, Takeo Kanade
The Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh PA 15213

ABSTRACT

To perform planetary exploration without human supervision, a complete autonomous robot must be able to model its environment and locate itself while exploring its surroundings. To this end, estimating motion between sensor views and merging the views into a coherent map are two important problems. In this paper, we present a 3-D perception system for building a geometrical description of rugged terrain environments from range data. We propose an intermediate representation consisting of an elevation map that includes an explicit representation of uncertainty and labeling of the occluded regions. We present an algorithm, called the *Locus* method, to convert range image to an elevation map. An uncertainty model based on this algorithm is developed. Based on this elevation map and the *Locus* method, we propose a terrain matching algorithm which does not assume any correspondences between range images. The algorithm consists of two stages: First, a feature-based matching algorithm computes an initial transform. Second, an iconic terrain matching algorithm that minimizes the correlation between two range images is applied to merge multiple range images into a uniform representation. We present terrain modeling results on real range images of rugged terrain.

1 INTRODUCTION

Exploration of the rugged terrain of Mars without human supervision requires a vehicle capable of both navigation and sampling. To navigate it must perceive its environment, plan a path through a model of the environment and maintain knowledge of its position. These requirements necessitate a system of perception that can build and maintain terrain maps which include information such as geometry and soil type. Building detailed 3-D geometrical descriptions is a critical technique for safe navigation and sampling in the unstructured rugged environment of Mars. A legged locomotor, such as the AMBLER [2] (shown in Figure 1), is dependent upon accurate 3-D knowledge of the terrain for reliable foot placement and leg recovery. Therefore, 3-D perception is a necessary component of the AMBLER system. In this paper, we discuss terrain modeling and representation issues for rugged terrain. We also present 3-D perception algorithms to construct detailed local terrain maps from a laser rangefinder. These algorithms are a fundamental part of the perception system for the AMBLER.

Elevation maps have been reported to be an effective representation method: A cartesian elevation map (CEM) representation is used for cross-country navigation [8]. Asada [1] used elevation map for fusing range and intensity data in a structured environment.

In this paper, the *Locus* Method is introduced for building iconic representations (elevation maps) which are applied to modeling rugged terrain. The *Locus* Method uses a model of the sensor and provides interpolation at arbitrary resolution without making any assumptions on the terrain shape other than the continuity of the surface. In order to take into account uncertainty inherent in sensor measurements, an uncertainty model for an iconic representation (elevation map) is proposed.

---

[1] This research was sponsored by NASA under Contract NAGW-1175. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies.

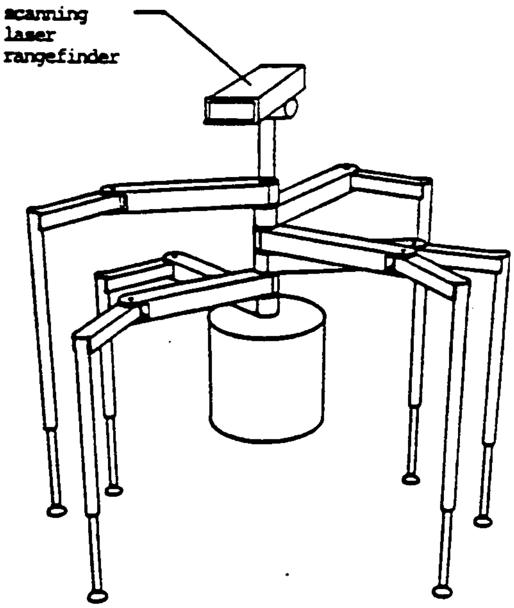


Figure 1: Sketch of Ambler

Feature-based matching methods, which compute the motion by finding correspondences between features, have been successful in structured environment (e.g., indoor navigation) [10,15,18,7,1,24]. These motion estimation methods, however, have limited successes for outdoor navigation , since extracting reliable features and finding correspondences are more difficult in rugged terrain environment. Recent work [23] introduces a method, based on a smooth surface assumption, for determining observer motion from sparse range data without assuming correspondence. Smooth surface assumption, however, may not be applicable for rugged terrain.

In this paper, we introduce an iconic matching method to optimally compute the motion which does not require any correspondences or smooth surface assumptions. The iconic method presented in this paper work directly on the two sets of data points, P^1 and P^2 by minimizing a cost function of the form $F(T(P^2), P^1)$ where $T(P^2)$ is the set of points from view 2 transformed by a displacement T. The cost is designed so that its minimum corresponds to a "best" estimate of T in some sense. The minimization of F leads to an iterative gradient-like algorithm. Although less popular, iconic techniques have been successfully applied to incremental depth estimation [18,17] and map matching [23,9]. Feature-based matching approach is used to compute an initial estimation of the motion which is important for fast convergence in our gradient descent optimization technique. The high curvature points are viewpoint-independent features that can be used for matching. We extract the high curvature points from both images of principal curvature. We group the extracted points into regions, then classify each region as point feature, line, or region, according to its size, elongation, and curvature distribution. We find the corresponding regions and compute the motion.

There are many sensing strategies for perception for mobile robots, including single camera systems, sonars, passive stereo, and laser range finders. In this paper, we focus on perception algorithms for range sensors that provide 3-D data directly by active sensing. Using such sensors has the advantage of eliminating the calibration problems and computational costs inherent in passive techniques such as stereo.

In this paper, we first describe the range sensor that we used in this work. Even though we tested our algorithm on one specific range sensor, we believe that the sensor characteristics are fairly typical of a wide range of sensors [4]. We explain the *Locus* Method which converts range images to elevation maps, and introduce uncertainty model for elevation maps taking into account the shape of sensor noise and the characteristics of the *Locus* Method. We demonstrate the performance of this algorithm by comparing it with traditional algorithms on test range images with different noise levels. We examine a couple of applications of elevation maps. Then, we introduce two matching algorithms: iconic and feature-based matching algorithms. These matching algorithms are tested and demonstrated on real range images of outdoor scene and synthetic range images.

2 ACTIVE RANGE SENSING

The basic principle of active sensing techniques is to observe the reflection of a reference signal (sonar, laser, radar..etc.) produced by an object in the environment in order to compute the distance between the sensor and that object.

Active sensors are attractive to mobile robots researchers for two main reasons: first, they provide range data without the computation overhead associated with conventional passive techniques such as stereo vision, which is important in time critical applications such as obstacle detection. Second, it is largely insensitive to outside illumination conditions, simplifying considerably the image analysis problem. This is especially important for images of outdoor scenes in which illumination cannot be controlled or predicted. In addition, active range finding technology has developed to the extent that makes it realistic to consider it as part of practical mobile robot implementations in the short term [4].

The range sensor we used is a time-of-flight laser range finder developed by the Environmental Research Institute of Michigan (ERIM). The basic principle of the sensor is to measure the difference of phase between a laser beam and its reflection from the scene [25]. A two-mirror scanning system allows the beam to be directed anywhere within a 30° × 80° field of view. The data produced by the ERIM sensor is a 64 x 256 range image, the range is coded on eight bits from zero to 64 feet, which corresponds to a range resolution of three inches. All measurements are all relative since the sensor measures differences of phase. That is, a range value is known *modulo* 64 feet. We have adjusted the sensor so that the range value 0 corresponds to the mirrors for all the images presented in this report. A first order approximation of the standard deviation of the range noise, σ is given by [13]:

$$\sigma \propto \frac{D^2}{\cos\theta} \tag{2.1}$$

The proportionality factor in this equation depends on the characteristics of the laser transmitter, the outside illumination, and the reflectance ρ of the surface which is assumed constant across the footprint in this first order approximation. Figure 2 shows a range image of an outdoor scene.

The position of a point in a given coordinate system can be derived from the measured range and the direction of the beam at that point. The cartesian coordinates of a point measured by the range sensor are given by the equations[2]:

$$\begin{aligned} x &= D\sin\theta \qquad (2.2) \\ y &= D\cos\phi\cos\theta \\ z &= D\sin\phi\cos\theta \end{aligned}$$

where ϕ and θ are the vertical and horizontal scanning angles of the beam direction corresponding to row and column position in the image. Figure 3 shows an overhead view of the scene of Figure 2, the coordinates of the points are computed using Eq. (2.3).

---

[2] Note that the reference coordinate system is not the same as in [12] for consistency reasons

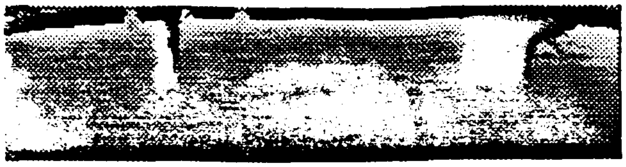


Figure 2: Range image

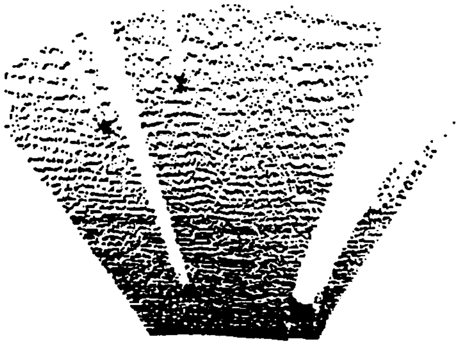


Figure 3: Overhead view

3 TERRAIN REPRESENTATIONS

The main task of 3-D vision in a mobile robot system is to provide sufficient information to the path planner so that the vehicle can be safely steered through its environment. In the case of outdoor navigation, the task is to convert a range image into a representation of the terrain. We use the word "terrain" in a very loose sense in that we mean both the ground surface and the objects that may appear in natural environments (*e.g.* rocks). In this Section we discuss the techniques that we have implemented for Mars Rover systems. We first introduce the concept of the elevation map as a basis for terrain representations and its relationship with different path planning techniques.

3.1 The elevation map as the data structure for terrain representation

Even though the format of the range data is an image, this may not be the most suitable structuring of the data for extracting information. For example, a standard representation in 3-D vision for manipulation is to view a range image as a set of data points measured on a surface of the equation $z = f(x,y)$ where the $x-$ and $y-$axes are parallel to the axis of the image and z is the measured depth. This choice of axis is natural since the image plane is usually parallel to the plane of the scene. In our case, however, the "natural" reference plane is not the image plane but is the ground plane. In this context, "ground plane" refers to a plane that is horizontal with respect to the vehicle or to the gravity vector. The representation $z = f(x,y)$ is then the usual concept of an elevation map. To transform the data points into an elevation map is useful only if one has a way to access them. The most common approach is to discretize the (x,y) plane into a grid. Each grid cell (x_i,y_i) is the trace of a vertical column in space, its *field*. All the data that falls within a cell's field is stored in that cell.

Although the elevation map is a natural concept for terrain representations, it exhibits a number of problems due to the conversion of a regularly sampled image to a different reference plane [13]. Although we propose solutions to these problems in Section 3.1, it is important to keep them in mind while we investigate other terrain representations. The first problem is the sampling problem: Since we perform some kind of image warping, the distribution of data points in the elevation map is not uniform, and as a result conventional image processing algorithms cannot be applied directly to the map. There are two ways to get around the sampling problem: We can either use a base structure that is not a regularly spaced grid, such as a Delaunay triangulation of the data points [19], or we can interpolate between data points to build a dense elevation map. The former solution is not very practical because of the complex algorithms required to access data points and their neighborhoods. We describe an implementation of the latter approach in Section 3.1. A second problem with elevation maps is the representation of the range shadows created by some objects. Since no information is available within the shadowed regions of the map, we must represent them separately so that no interpolation takes place across them and no "phantom" features are reported to the path planner. Finally, we have to convert the noise on the original measurements into a measure of uncertainty on the z value at each grid point (x, y). This conversion is difficult due to the fact that the sensor's uncertainty is most naturally represented with respect to the direction of measurement and therefore spreads across a whole region in the elevation map.

The elevation map derived directly from the sensor is sparse and noisy, especially at greater distances from the sensor. Many applications, however, need a dense and accurate high resolution map. One way to derive such a map is to interpolate between the data points using some mathematical approximation of the surface between data points. The models that can be used include linear, quadratic, or bicubic surfaces [19]. Another approach is to fit a surface globally under some smoothness assumptions. This approach includes the family of regularization algorithms [6].

Two problems arise with both interpolation approaches: They make priori assumptions on the local shape of the terrain which may not be valid (*e.g.* in the case of very rough terrain), and they do not take into account the image formation process since they are generic techniques independent of the origin of the data. In addition, the interpolation approaches depend heavily on the resolution and position of the reference grid. For example, they cannot compute an estimate of the elevation at an (x, y) position that is not a grid point without resampling the grid. We propose an alternative, the *locus* algorithm [13], that uses a model of the sensor and provides interpolation at arbitrary resolution without making any assumptions on the terrain shape other than the continuity of the surface.

3.1.1 The *Locus* algorithm for the optimal interpolation of terrain maps

The problem of finding the elevation z of a point (x,y) is equivalent to computing the intersection of the surface observed by the sensor and the vertical line passing through (x, y). The basic idea of the locus algorithm is to convert the latter formulation into a problem in image space (Figure 4). A vertical line is a curve in image space, the *locus*, whose equation as a function of ϕ is:

$$D = D_l(\phi) = \sqrt{\frac{y^2}{\cos^2\phi} + x^2} \tag{3.3}$$

$$\theta = \theta_l(\phi) = \arctan\frac{x\cos\phi}{y} \tag{3.4}$$

where ϕ, θ, and D are defined as in Section 2. Equation (3.4) was derived by inverting Equation (2.3), and assuming x and y constant. Similarly, the range image can be viewed as a surface $D = I(\phi, \theta)$ in ϕ, θ, D space. The problem is then to find the intersection, if it exists, between a curve parameterized by ϕ and a discrete surface. Since the surface is known only from a sample of data, the intersection cannot be computed analytically. Instead, we have to search along the curve for the intersection point. The search proceeds in two stages: We first locate the two scanlines of the range image, ϕ_1 and ϕ_2, between which the intersection must be located, that is the two consecutive scanlines such that, $Diff(\phi_1) = D_l(\phi_1) - I(\phi_1, \hat{\theta}_l(\phi_1))$ and $Diff(\phi_2) = D_l(\phi_1) - I(\phi_2, \hat{\theta}_l(\phi_2))$ have opposite signs, where $\hat{\theta}_l(\phi)$ is the image column that is the closest to $\theta_l(\phi)$. We then apply a binary search between ϕ_1 and ϕ_2. The search stops when the difference between the two angles ϕ_n and ϕ_{n+1}, where $Diff(\phi_n)$ and $Diff(\phi_{n+1})$ have opposite signs, is lower than a threshold ϵ. Since there are no pixels between ϕ_1 and ϕ_2, we have to perform a local quadratic interpolation of the image in order to compute $\theta_l(\phi)$ and $D_l(\phi)$ for $\phi_1 < \phi < \phi_2$. The control points for the interpolation are the four pixels that surround the intersection point. The final result is a value ϕ that is converted to an elevation value by applying Equation (2.3) to $\phi, \theta_l(\phi), D_l(\phi)$. The resolution of the elevation is controlled by the choice of the parameter ϵ.

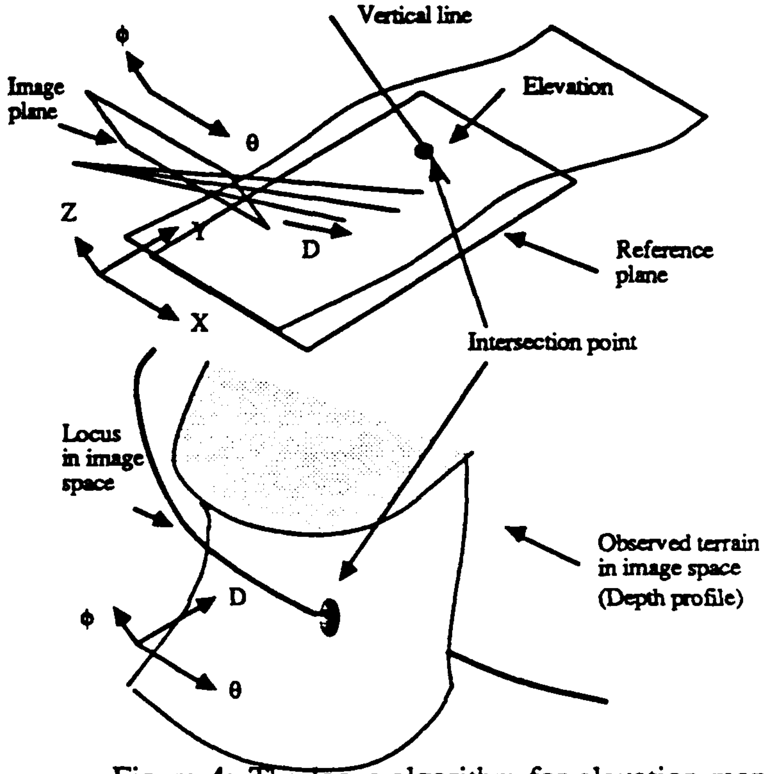


Figure 4: The locus algorithm for elevation maps

The locus algorithm enables us to evaluate the elevation at any point since we do not assume the existence of a grid. The uncertainty in the elevation can be approximated by a Gaussian distribution [13].

Figure 5 shows the result of applying the locus algorithm on range images of uneven terrain, in this case a construction site. The Figure shows the original range image and the elevation map displayed as an isoplot surface. The centers of the grid cells are ten centimeters apart in the (x, y) plane.

3.1.2 Generalizing the locus algorithm

We can generalize the locus algorithm from the case of a vertical line to the case of a general line in space. This generalization allows us to build maps using any reference plane instead of being restricted to the (x, y) plane. This is important when, for example, the sensor's (x, y) plane is not orthogonal to the gravity vector. A line in space is defined by a point $u = [u_x, u_y, u_z]^t$, and a unit vector $v = [v_x, v_y, v_z]^t$. Such a line is parameterized in λ by the relation $p = u + \lambda v$ if p is a point on the line. A general line is still a curve in image space that can be parameterized in ϕ if we assume that the line is not parallel to the (x, y) plane. The equation of the

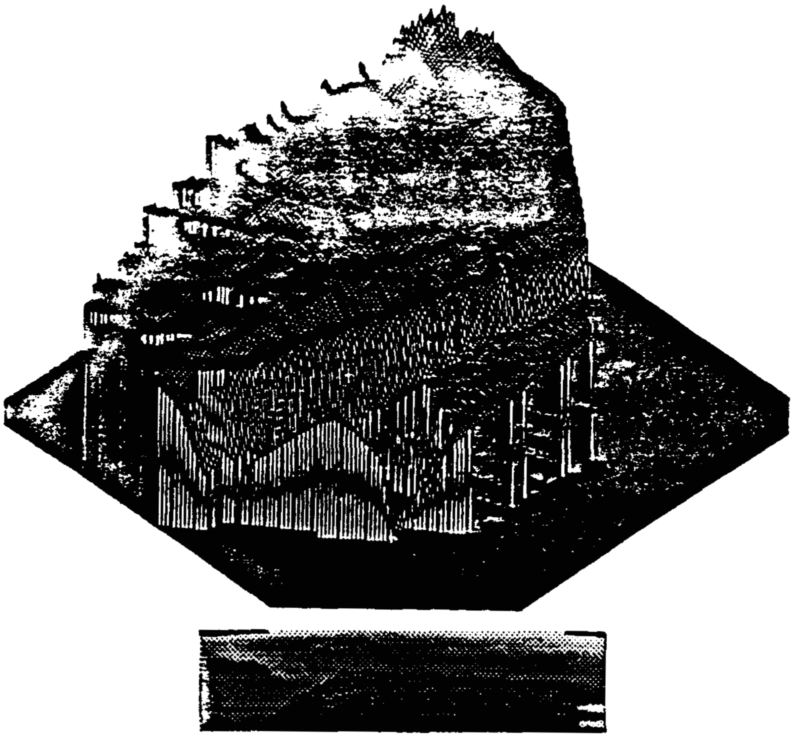


Figure 5: The locus algorithm on range images

curve becomes:

$$\begin{aligned} D_l(\phi) &= \sqrt{(v_x\lambda(\phi)+u_x)^2+(v_y\lambda(\phi)+u_y)^2+(v_z\lambda(\phi)+u_z)^2} \\ \theta_l(\phi) &= \arcsin\frac{v_x\lambda(\phi)+u_x}{D} \\ \lambda(\phi) &= \frac{u_y\tan\phi-u_z}{v_z-v_y\tan\phi} \end{aligned} \tag{3.5}$$

We can then compute the intersection between the curve and the image surface by using the same algorithm as before except that we have to use Equation (3.5) instead of Equation (3.4).

The representation of the line by the pair (u,v) is not optimal since it uses six parameters while only four parameters are needed to represent a line in space. For example, this can be troublesome if we want to compute the Jacobian of the intersection point with respect to the parameters of the line. A better alternative [14] is to represent the line by its slopes in x and y and by its intersection with the plane $z=0$ (See [21] for a complete survey of 3-D line representations). The equation of the line then becomes:

$$\begin{aligned} x &= az+p \\ y &= bz+q \end{aligned} \tag{3.6}$$

We can still use Equation (3.5) to compute the locus because we can switch between the (a,b,p,q) and (u,v) representations [13].

3.1.3 Evaluating the locus algorithm

We evaluate the locus algorithm by comparing its performance with the traditional interpolation algorithms on a set of synthesized range images of simple scenes. The simplest scenes are planes at various orientations. Furthermore, we add some range noise in order to evaluate the robustness of the approach in the presence of noise. The performances of the algorithms are evaluated by using the mean square error:

$$E = \frac{\sum_{i=1}^{N}(h_i - \tilde{h}_i)^2}{N} \tag{3.7}$$

where h_i is the true elevation value and $\tilde{h}_i$ is the estimated elevation. Figure 6 plots E for the locus algorithm and the naive interpolation as a function of the slope of the observed plane and the noise level. This result shows that the locus algorithm is more stable with respect to surface orientation and noise level than the other algorithm. This is due to the fact that we perform the interpolation in image space instead of first converting the data points into the elevation map.

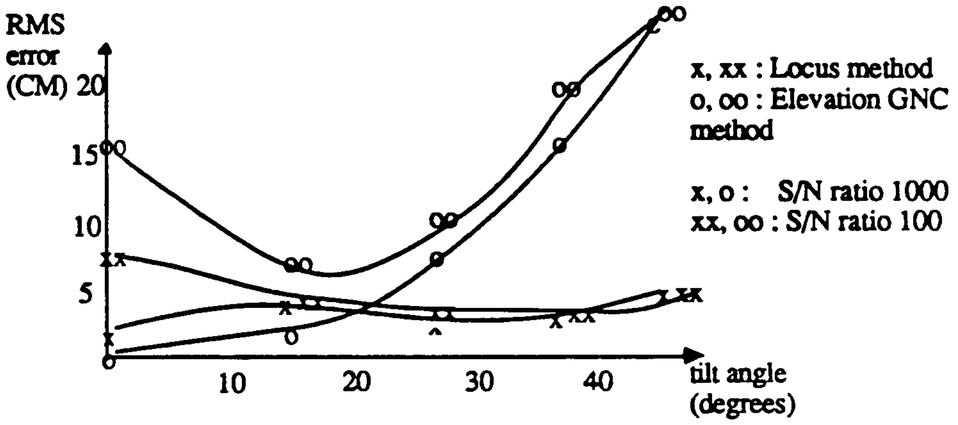


Figure 6: Evaluation of the locus algorithm on synthesized images

3.1.4 Detecting the range shadows

As we pointed out in Section 3.1, the terrain may exhibit range shadows in the elevation map. It is important to identify the shadow regions because the terrain may have any shape within the boundaries of the shadows, whereas the surface would be smoothly interpolated if we applied the locus algorithm directly in those areas. This may result in dangerous situations for the robot if a path crosses one of the range shadows. A simple idea would be to detect empty regions in the raw elevation map, which are the projection of images in the map without any interpolation. This approach does not work because the size of the shadow regions may be on the order of the average distance between data points. This is especially true for shadows that are at some distance from the sensor in which case the distribution of data points is very sparse. It is possible to modify the standard locus algorithm so that it takes into account the shadow areas. The basic idea is that a range shadow corresponds to a strong occluding edge in the image. An (x, y) location in the map is in a shadow area if its locus intersects the image at a pixel that lies on such an edge (Figure 7).

We implement this idea by first detecting the edges in the range image by using a standard technique, the GNC algorithm [6]. We chose this algorithm because it allows us to vary the sensitivity of the edge detector across the image, and because it performs some smoothing of the image as a side effect. When we apply the locus algorithm we can then record the fact that the locus of a given location intersects the image at an edge pixel. Such map locations are grouped into regions that are the reported range shadows. Figure 8 shows an

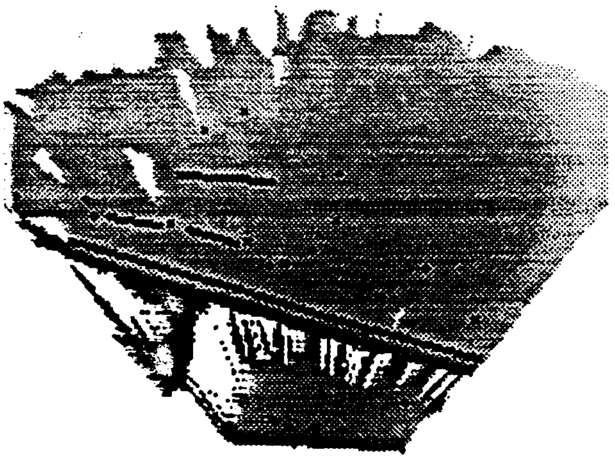


Figure 11: Local features from a high resolution elevation map

elevation map. The Figure shows that while some features correspond merely to local extrema of the surface, some such as the edges of the deep gully are characteristic features of the scene. This type of feature extraction plays an important role in Section 4 for combining multiple maps computed by the locus algorithm.

4 COMBINING MULTIPLE TERRAIN MAPS

We have so far addressed the problem of building a representation of the environment from sensor data collected at one fixed location. In the case of mobile robots, however, we have to deal with a stream of images taken along the vehicle's path. First of all, merging representations from successive viewpoints will produce a map with more information and better resolution than any of the individual maps. Another reason why merging maps increases the resolution of the resulting representation concerns the fact that the resolution of an elevation map is significantly better at close range. By merging maps, we can increase the resolution of the parts of the elevation map that were originally measured at a distance from the vehicle.

The second motivation for merging maps is that the position of the vehicle at any given time is uncertain. Even when using expensive positioning systems, we have to assume that the robot's idea of its position in the world will degrade in the course of a long mission. One way to solve this problem is to compute the position with respect to features observed in the world instead of a fixed coordinate system [22,18]. That requires the identification and fusion of common features between successive observations in order to estimate the displacement of the vehicle.

In the terrain matching problem, there are two possible approaches: feature-based or iconic matching. In feature-based matching, we first have to extract two sets of features (F_i^1) and (F_j^2) from the two views to be matched, and to find correspondences between features, $(F_{i_k}^1, F_{j_k}^2)$ that are globally consistent. We can then compute the displacement between the two views from the parameters of the features, and finally merge them into one common map. Although this is the standard approach to object recognition problems [5], it has also been widely used for map matching for mobile robots [10,15,18,7,1,24]. In contrast, iconic approaches work directly on the two sets of data points, P^1 and P^2 by minimizing a cost function of the form $F(T(P^2), P^1)$ where $T(P^2)$ is the set of points from view 2 transformed by a displacement T. The cost is designed so that its minimum corresponds to a "best" estimate of T in some sense. The minimization of F leads to an iterative gradient-like algorithm. Although less popular, iconic techniques have been successfully applied to incremental depth estimation [18,17] and map matching [23,9]. We describe in detail the feature-based and iconic stages in the next three sections.

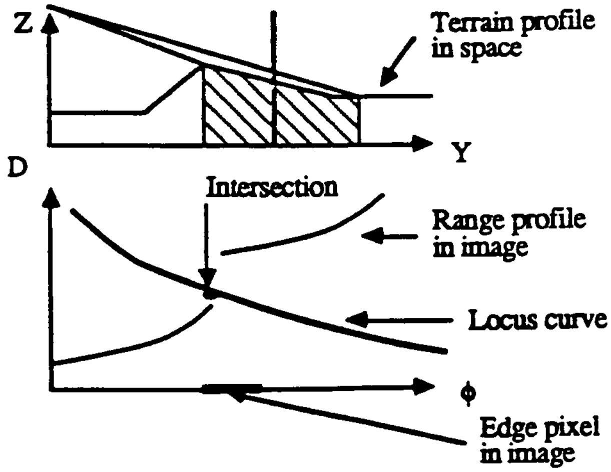


Figure 7: Detecting range shadows

overhead view of an elevation map computed by the locus algorithm, the white points are the shadow points, the gray level of the other points is proportional to their uncertainty as computed in the previous Section.

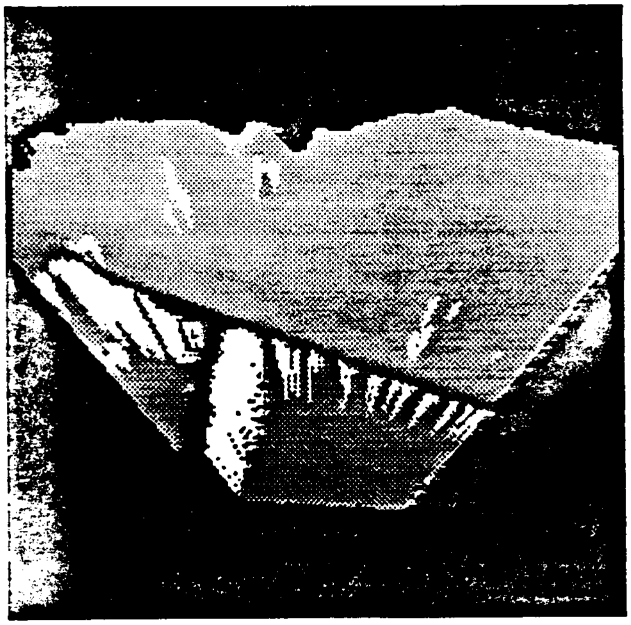


Figure 8: Shadow regions in an elevation map

3.1.5 An application: footfall selection for the Ambler

The purpose of using the locus algorithm for building terrain is to provide high resolution elevation data. As an example of an application in which such a resolution is needed, we briefly describe in this Section the problem of perception for a legged vehicle [16]. One of the main responsibilities of perception for a legged vehicle is to provide a terrain description that enables the system to determine whether a given foot placement, or *footfall*, is safe. In addition, we consider the case of locomotion on very rugged terrain such as the surface of Mars.

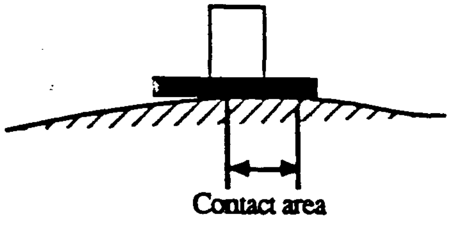


Figure 9: Footfall support area

A foot is modeled by a flat disk of diameter 30 cms. The basic criterion for footfall selection is to select a footfall area with the maximum support area which is defined as the contact area between the foot and the terrain as shown in Figure 9. Another constraint for footfall selection is that the amount of energy necessary to penetrate the ground in order to achieve sufficient support area must be minimized. The energy is proportional to the depth of the foot in the ground. The support area is estimated by counting the number of map points within the circumference of the disk that are above the plane of the foot. This is where the resolution requirement originates because the computation of the support area makes sense only if the resolution of the map is significantly smaller than the diameter of the foot. Given a minimum allowed support area, S_{min}, and the high resolution terrain map, we can find the optimal footfall position within a given terrain area: First, we want to find possible flat areas by computing surface normals for each footfall area in a specified footfall selection area. Footfalls with a high surface normal are eliminated. The surface normal analysis, however, will not be sufficient for optimal footfall selection. Second, the support area is computed for the remaining positions. The optimal footfall position is the one for which the maximum elevation, h_{opt} that realizes the minimum support area S_{min} is the maximum across the set of possible footfall positions. Figure 10 shows a plot of the surface area with respect to the elevation from which h_{min} can be computed.

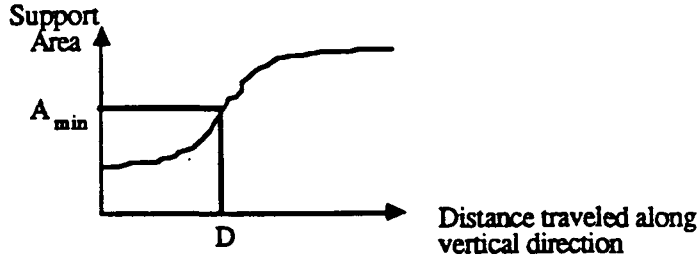


Figure 10: Support area versus elevation

3.1.6 Extracting local features from an elevation map

The high resolution map enables us to extract very local features, such as points of high surface curvature. The local features that we extract are based on the magnitude of the two principal curvatures of the terrain surface. The curvatures are computed as in [20] by first smoothing the map, and then computing the derivatives of the surface for solving the first fundamental form. Points of high curvature correspond to edges of the terrain, such as the edges of a valley, or to sharp terrain features such as hills, or holes. In any case, the high curvature points are viewpoint-independent features that can be used for matching. We extract the high curvature points from both images of principal curvature. We group the extracted points into regions, then classify each region as point feature, line, or region, according to its size, elongation, and curvature distribution. Figure 11 shows the three types of local features detected on the map of Figure 8 superimposed in black over the original

4.1 Feature-based matching

Let F_i^1 and F_j^2 be two sets of features extracted from two images, I_1 and I_2. We want to find a transformation $\hat{T}$ and a set of pairs $C_k = (F_{i_k}^1, F_{j_k}^2)$ such that $F_{j_k}^2 \approx \hat{T}(F_{i_k}^1)$, where $T(F)$ denotes the transformed by T of a feature F. The features can be any of those discussed in the previous Sections: points or lines from the local feature extractor, obstacles represented by a ground polygon, or terrain patches represented by their surface equation and their polygonal boundaries.

We obtain a set of possible matchings, each of which is a set of pairs $S = (F_{i_k}^1, F_{j_k}^2)_k$ between the two sets of features. Since we evaluated T simply by combining components in the course of the search, we have to evaluate T for each S in order to get an accurate estimate. T is estimated by minimizing an error function of the form:

$$E = \sum_k d(F_{i_k}^1 - T(F_{i_k}^2)) \tag{4.8}$$

The distance $d(.)$ used in Equation (4.8) depends on the type of the features involved. The matching S that realizes the minimum E is reported as the final match between the two maps while the corresponding displacement $\hat{T}$ is reported as the best estimate of the displacement between the two maps.

Feature-based matching is applied to estimate the displacement for the iconic matching of high resolution maps. The primitives used for the matching are the high curvature points and lines described in Section 3.1.6. The initial matches are based on the similarity of the length of the lines and the similarity of the curvature strength of the points. The search among candidate matches proceeds as described in Section 4.1. Since we have dense elevation at our disposal in this case, we can evaluate a candidate displacement over the entire map by summing up the squared differences between points in one map and points in the transformed map. Figure 12 shows the result of the feature matching in a case in which the maps are separated by a very large displacement. The top image shows the superimposition of the contours and features of the two maps using the estimated displacement , while the bottom image shows the correspondences between the point and line features in the two maps. The lower map is transformed by T with respect to the lower right map. The lower left display shows the area that is common between the two maps after the displacement.

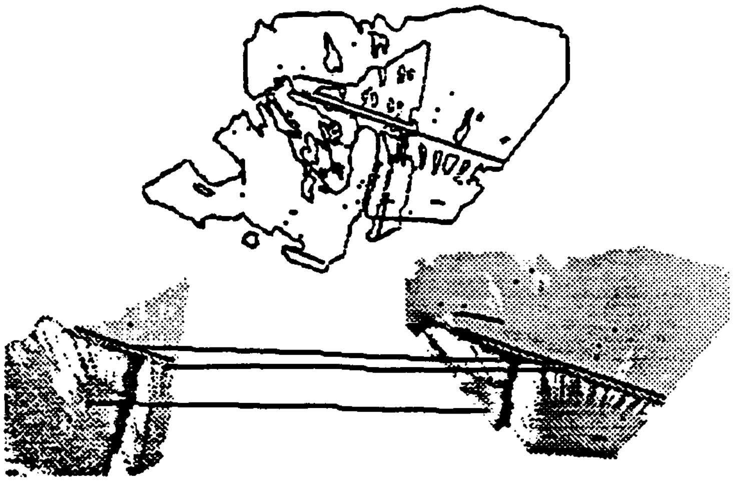


Figure 12: Matching maps using local features

4.2 Iconic matching from elevation maps

The general idea of the iconic matching algorithm is to find the displacement T between two elevation maps from two different range images that minimizes an error function computed over the entire combined elevation map. The error function E measures how well the first map and the transformed of the second map by T do agree. The easiest formulation for E is the sum of the squared differences between the elevation at a location in the first map and the elevation at the same location computed from the second map using T. To be consistent with the earlier formulation of the locus algorithm, the elevation at any point of the first map is actually the intersection of a line containing this point with the range image. We need some additional notations to formally define E: R and t denote the rotation and translation parts of T respectively, $f_i(u,v)$ is the function that maps a line in space described by a point and a unit vector to a point in by the generalized locus algorithm of Section 3.1.2 applied to image i. We have then:

$$E = \sum \|f_1(u,v) - g(u,v,T)\|^2 \tag{4.9}$$

where $g(u,v,T)$ is the intersection of the transformed of the line (u,v) by T with image 2 expressed in the coordinate system of image 1 (Figure 13). The summation in Equation (4.9) is taken over all the locations (u,v) in the first map where both $f_1(u,v)$ and $g(u,v,T)$ are defined. The lines (u,v) in the first map are parallel to the z-axis. In other words:

$$g(u,v,T) = T^{-1}(f_2(u',v')) = R'f_2(u',v') + t' \tag{4.10}$$

where $T^{-1} = (R',t') = (R^{-1}, -R^{-1}t)$ is the inverse transformation of T, and $(u',v') = (Ru + t, Rv)$ is the transformed of the line (u,v). This Equation demonstrates one of the reasons why the locus algorithm is powerful: in order to compute $f_2(Ru + t, Rv)$ we can apply directly the locus algorithm, whereas we would have to do some interpolation or resampling if we were using conventional grid-based techniques. We can also at this point fully justify the formulation of the generalized locus algorithm in Section 3.1.2: The transformed line (u',v') can be anywhere in space in the coordinate system of image 2, even though the original line (u,v) is parallel to the z-axis, necessitating the generalized locus algorithm to compute $f_2(u',v')$.

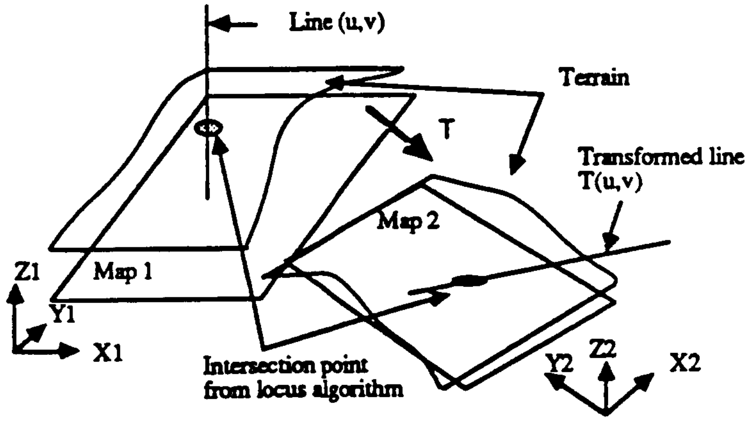


Figure 13: Principle of the iconic matching algorithm

We now have to find the displacement T for which E is minimum. If $\nu = [\alpha, \beta, \gamma, t_x, t_y, t_z]^t$ is the 6-vector of parameters of T, where the first three components are the rotation angles and the last three are the components of the translation vector, then E reaches a minimum when:

$$\frac{\partial E}{\partial \nu} = 0 \tag{4.11}$$

Assuming an initial estimate T_0, such a minimum can be found by an iterative gradient descent of the form:

$$\nu^{i+1} = \nu^i + k\frac{\partial E}{\partial \nu}(\nu^i) \tag{4.12}$$

where ν^i is the estimate of ν at iteration i. From Equation (4.9), the derivative of E can be computed by:

$$\frac{\partial E}{\partial \nu} = -2\sum(f_1(u,v) - g(u,v,T))\frac{\partial g}{\partial \nu}(u,v,T) \tag{4.13}$$

From Equation (4.10), we get the derivative of g:

$$\frac{\partial g}{\partial \nu}(u,v,T) = R'\frac{\partial f_2}{\partial \nu}(u',v') + \frac{\partial R'}{\partial \nu}f_2(u',v') + \frac{\partial t'}{\partial \nu} \tag{4.14}$$

The derivatives appearing in the last two components in Equation (4.14) are the derivatives of the transformation with respect to its parameters which can be computed analytically. The last step to compute the derivative of $g(u,v,T)$ is therefore to compute the derivative of $f_2(u',v')$ with respect to ν. We could write the derivative with respect to each component ν_i of ν by applying the chain rule directly:

$$\frac{\partial f_2}{\partial \nu_i}(u',v') = \frac{\partial f_2}{\partial u}\frac{\partial u'}{\partial \nu_i} + \frac{\partial f_2}{\partial v}\frac{\partial v'}{\partial \nu_i} \tag{4.15}$$

Equation (4.15) leads however to instabilities in the gradient algorithm because, as we pointed out in Section 3.1.2, the (u,v) representation is an ambiguous representation of lines in space. We need to use a non ambiguous representation in order to correctly compute the derivative. Since we can use interchangeably the (u,v) representation and the unambiguous (a,b,p,q) representation, we can correctly compute the derivative [13].

In the actual implementation of the matching algorithm, the points at which the elevation is computed in the first map are distributed on a square grid of ten centimeters resolution. The lines (u,v) are therefore vertical and pass through the centers of the grid cells. E is normalized by the number of points since the overlap region between the two maps is not known in advance. We first compute the $f_1(u,v)$ for the entire grid for image 1, and then apply directly the gradient descent algorithm described above. The iterations stop either when the variation of error ΔE is small enough, or when E itself is small enough. Since the matching is computationally expensive, we compute E over an eight by eight meter window in the first image. The last test ensures that we do not keep iterating if the error is smaller than what can be reasonably achieved given the characteristics of the sensor. Figure 14 shows the result of combining three high resolution elevation maps. The displacements between maps are computed using the iconic matching algorithm. The maps are actually combined by replacing the elevation $f_1(u,v)$ by the combination:

$$\frac{\sigma_2 f_1 + \sigma_1 f_2}{\sigma_1 + \sigma_2} \tag{4.16}$$

where σ_1 and σ_2 are the uncertainty values. The resulting mean error in elevation is lower than ten centimeters.

We computed the initial T_0 by using the local feature matching of Section 4.1. This estimate is sufficient to ensure the convergence to the true value. This is important because the gradient descent algorithm converges towards a local minimum, and it is therefore important to show that T_0 is close to the minimum. Table 1 shows the converged and initial values of the ν_i's by the iterative matching algorithm for matching two maps separated by a very large displacement as shown in Figure 12. In this experiment, the initial estimate is obtained by feature-based matching method described in Section 4.1 and the gradient descent algorithm needed approximately 30 iterations to converge to the minimum. Since the ground truth cannot be obtained, we do not know the absolute accuracy of those motion parameters. However, from the fact that the elevation RMS error is about 6 cm, the iconic matching method combined with feature-based matching is very satisfactory for our application in rugged terrain.

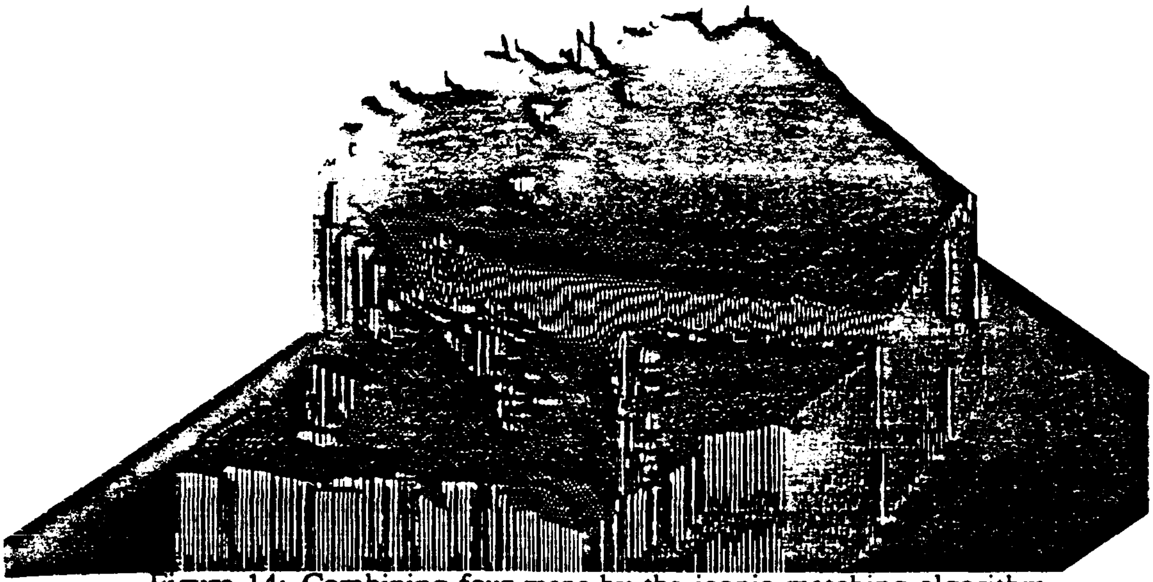


Figure 14: Combining four maps by the iconic matching algorithm

	Initial values	Converged values
Translation (cm)	(-108.9, -125.0, 0.0)	(-191.26, -119.265, -19.56
Rotation (degree)	(0.0, 0.0, 29.5917)	(1.93, -0.51566, 31.114)
RMS Error (cm/pixel)	81.03	5.92

Table 1: Results for computing motion parameters

5 CONCLUSION

In this paper, we have introduced a new method (*Locus*) to convert range images to elevation maps. This method is used to develop an uncertainty model and an algorithm for representing vertical objects in elevation maps. We have demonstrated the applicability of elevation maps to the footfall selection of a legged vehicle. An iterative iconic matching technique without assuming any correspondence or smooth surface assumptions is developed to compute the optimal motion. This motion estimate is then used to obtain a composite terrain map by merging multiple range images. We have demonstrated these algorithms on real range images of outdoor scenes. The results suggest that elevation maps are in general an appropriate terrain representation and our iconic matching method is useful for rugged terrain environments.

Many issues still remain to be investigated. First of all, we must define a uniform way of representing and combining the uncertainties in the terrain maps. Currently, the uncertainty models depend heavily on the type of sensor used and on the level at which the terrain is represented. Furthermore, the displacements between terrain maps are known only up to a certain level of uncertainty. This level of uncertainty must be evaluated and updated through the matching of maps, whether iconic or feature-based. We have tackled the terrain representation problems mainly from a geometrical point of view.

A natural extension of this work is to use the 3-D terrain representations to identify known objects in the scene. Another application along these lines is to use the terrain maps to identify objects of interest, such as terrain regions for sampling tasks for a planetary explorer [16]. Although we have performed some preliminary experiments in that respect [11,2], extracting semantic information from terrain representations remains a major research area for outdoor mobile robots.

References

[1] M. Asada. Building a 3-D World Model for a Mobile Robot from Sensory Data. In *Proc. IEEE Robotics and Automation*, Philadelphia, 1988.

[2] J. Bares and W. Whittaker. Configuration of an autonomous robot for Mars exploration. In *World Robotics Conference, Society of Mechanical Engineers*, To appear, 1988.

[3] J. Bares and W. Whittaker. Configuration of an Autonomous Robot for Mars Exploration. In *Proc. World Conference on Robotics*, 1988.

[4] P. Besl. *Range Imaging Sensors*. Technical Report GMR-6090, General Motors Research Lab, Warren, MI, March 1988.

[5] P. J. Besl and R. C. Jain. Three-dimensional Object Recognition. *ACM Comp. Surveys*, 17(1), march 1985.

[6] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.

[7] R. Brooks. Aspects of Mobile Robot Visual Map Making. In *Second International Robotics Research Symposium*, MIT press, 1985.

[8] M.J. Daily, J.G. Harris, and K. Reiser. An Operational Perception System for Cross-Country Navigation. In *Proc. Image Understanding Workshop*, Cambridge, 1988.

[9] A. Elfes. Sonar-Based Real-World Mapping and Navigation. *Journal of Robotics and Automation, Vol. 3*, 1987.

[10] O.D. Faugeras, N. Ayache, and B. Faverjon. Building Visual Maps by Combining Noisy Stereo Measurements. In *Proc. IEEE Conf. on Robotics and Automation*, 1986.

[11] M. Hebert and T. Kanade. 3-D Vision for Outdoor Navigation by an Autonomous Vehicle. In *Proc. Image Understanding Workshop*, Cambridge, 1988.

[12] M. Hebert and T. Kanade. First Results on Outdoor Scene Analysis. In *Proc. IEEE Robotics and Automation*, San Francisco, 1985.

[13] M. Hebert, T. Kanade, and I. Kweon. *3-D Vision Techniques for Autonomous Vehicles*. Technical Report CMU-RI-TR-88-12, The Robotics Institute, Carnegie-Mellon University, 1988.

[14] M.G. Kendall and P.A.P. Moran. *Geometrical Probabilities*. Hafner Publishers, New York, 1963.

[15] D. Kriegman, E. Triendl, and T.O. Binford. A Mobile Robot: Sensing, Planning and Locomotion. In *Proc. IEEE Conf. on Robotics and Automation*, 1987.

[16] I. Kweon. Modeling Rugged 3-D Terrain from Multiple Range Images for Use by Mobile Robots. 1988. PhD thesis proposal.

[17] B.D. Lucas. *Generalized Image Matching by the Method of Differences*. Technical Report CMU-CS-85-160, Carnegie-Mellon University, 1985.

[18] L. Matthies and S.A. Shafer. Error Modeling in Stereo Navigation. *Journal of Robotics and Automation, Vol. 3*, 1987.

[19] D.J. Orser and M. Roche. The Extraction of Topographic Features in Support of Autonomous Underwater Vehicle Navigation. In *Proc. Fifth International Symposium on Unmanned Untethered Submersible*, University of New Hampshire, 1987.

[20] J. Ponce and M. Brady. Toward a Surface Primal Sketch. In *IEEE International Conference on Robotics and Automation*, St Louis, 1985.

[21] K.S. Roberts. A New Representation for a Line. In *Proc. Computer Vision and Patter Recognition*, Ann Arbor, MI, 1988.

[22] R.C. Smith and P. Cheeseman. On the Representation and Estimation of Spatial Uncertainty. *International Journal of Robotics Research*, 1986.

[23] R. Szeliski. Estimating Motion from Sparse Range Data without Correspondance. In *International Conf. on Computer Vision*, Tarpon Springs, Florida, December 1988.

[24] C.E. Thorpe. *The CMU Rover and the FIDO Vision and Navigation System*. PhD thesis, Carnegie-mellon University, 1984.

[25] D. Zuk, F. Pont, R. Franklin, and V. Larrowe. *A System for Autonomous Land Navigation*. Technical Report IR-85-540, Environmental Research Institute of Michigan, Ann Arbor MI, 1985.